Transfer of tilted sample information in transmission electron microscopy

Inauguraldissertation

zur

Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel

von

Valerio Mariani
aus Monza, Italien

Basel, 2010

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät
auf Antrag von
Prof. Dr. Andreas Engel und Prof. Dr. Henning Stahlberg

Basel, den 26.5.2009

Prof. Dr. Eberhard Parlow
Dekan

creative commons

Attribution-Noncommercial-No Derivative Works 2.5 Switzerland

---

You are free:

to Share — to copy, distribute and transmit the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.
- Any of the above conditions can be waived if you get permission from the copyright holder.
- Nothing in this license impairs or restricts the author's moral rights.

Your fair dealing and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license) available in German:
http://creativecommons.org/licenses/by-nc-nd/2.5/ch/legalcode.de

Disclaimer:
The Commons Deed is not a license. It is simply a handy reference for understanding the Legal Code (the full license) — it is a human-readable expression of some of its key terms. Think of it as the user-friendly interface to the Legal Code beneath. This Deed itself has no legal value, and its contents do not appear in the actual license. Creative Commons is not a law firm and does not provide legal services. Distributing of, displaying of, or linking to this Commons Deed does not create an attorney-client relationship.

Quelle: http://creativecommons.org/licenses/by-nc-nd/2.5/ch/deed.en Datum: 3.4.2009

University Of Basel - M.E. Muller Institute

Transfer of tilted sample information in transmission electron microscopy

By Valerio Mariani

Andreas Engel

Henning Stahlberg

Contents

Introduction

When a transmission electron microscope is used in imaging mode, information from the sample function is not carried directly to the final observed micrograph, but it transformed during the imaging process in a way that is described by the so-called imaging function. When the sample is not tilted and no defocus gradient is present across its extent, the imaging function has a well-known and extensively studied form : the Contrast Transfer Function (CTF) (Reimer, 1997). Several electron microscopy techniques, however, require the sample to be tilted to fully explore its 3-dimensional structure. Only recently a rigorous mathematical description for the imaging process under these conditions, derived from physical first principles, has been made available: the Tilted Contrast Imaging Function (TCIF) (Philippsen et al., 2006). This manuscript discusses in depth the nature and the characteristics of the TCIF model. A robust and efficient software implementation is presented, developed with the context of the IPLT software development framework (Philippsen et al., 2007). Computer simulations are then used to qualitatively and quantitatively analyze features of experimental images, and to evalluate the performance of several available methods that are available to correct artifacts in images of titled samples.

The manuscript is divided in 3 parts. The first part, comprising Chapters 1 and 2 discusses the TCIF model in general without focusing on any specific electron microscopy technique. Chapter 1 consists of the draft of a journal article in a very advanced stage of preparation, almost ready for publication. Chapter

2 expands on the content of the previous chapter, and contains all the material that could not fit in the compact layout required for publication.

The second part focuses on the consequences of the TCIF model for the field of electron crystallography, and comprises Chapters 3 and 4. Chapter 3 is another draft of a future publication, although in this case at a much earlier stage of development. Chapter 4 presents additional material that might be included in the publication in the future, but has yet to be arranged in a suitable form.

Finally, the third part of the manuscript describes supplementary work that no direct connection with the subject of the previous chapters: algorithms for general image processing, contributions to the ongoing development of an electron crystallography data processing pipeline, and a proof-of-concept implementation of a multi-resolution modeling application in IPLT.

The presence of the two publication drafts cause stylistic inconsistencies within the manuscript. The inconsistencies involve graphical elements, as fonts, and also the numbering of figures and equations. All references within the paper drafts are internally consistent. Effort has been put into making all the cross-chapter references as clear as possible.

Chapter 1

Simulation and Correction of Tilted Weak-phase Samples

The next pages of this manuscript contain the draft of a journal article. The page, equation and figure numberings of the article have been preserved for an easier readibility.

Simulation and Correction of tilted weak-phase samples

Valerio Mariani, Andreas D. Schenk, Andreas Engel and Ansgar Philippsen

Maurice E. Müller Institute of Structural Biology, Basel

Corresponding Author: **ansgar.philippsen@unibas.ch**

Abbreviations and Symbols

TCIF Tilted Contrast Imaging Function

CTF Contrast Transfer Function

FT Fourier transform

PSF Point Spread Function

α tilt angle of specimen out of plane

β orientation of tilt axis within image plane, β=0 means tilt axis is along y-axis

$\mathbf{C_s}$ Spherical aberration constant of instrument

Δ Sampling distance in digitized image

λ Electron wavelength, calculated from acceleration voltage

N Number of pixels in digitized image

$\mathbf{z_0}$ Base defocus (defocus at center of specimen)

χ_0 Reference angle for astigmatism

χ Astigmatism angle

$\mathbf{z_a}$ Defocus difference due to astigmatism

$\mathbf{W_0}$ Scherzer formula at the base defocus

Q Transformation describing tilted contrast imaging

Å 10^{-10}m

Introduction

Modern high voltage electron microscopes produce images that represent a projection of the potential distribution of the sample on a 2D plane perpendicular to the electron beam (Lenz 1971), under the assumption that the sample is a weak phase object, which holds for most biological specimens. This allows electron microscopy to determine the structure of such specimens, provided that the gap between the 2-dimenionality of the projections and the 3-dimensionality of the structure can be bridged. This is achieved by combining information from projections recorded at different orientations, either stemming from random orientations in the same sample, or by collecting data at different tilts of the sample stage.

The information transfer in the imaging process can be described using phase contrast theory, given that the sample is limited in thickness, obeys the weak phase approximation, and a highly coherent microscope is used. For samples that are not tilted, where distance from focus is space invariant, this leads to a very convenient expression for the Contrast Transfer Function (CTF) (Reimer, 1997). For tilted samples, however, where distance from focus varies across the sample, a new fundamental optical problem is introduced and the classical CTF model must be abandoned, as we have demonstrated in a previously published rigorous mathematical treatment of the imaging process in tilted geometry (Philippsen et al., 2008B), which introduced the Tilted Contrast Imaging Function (TCIF). In this work, we move from the theoretical treatment of the TCIF into more applied and practical considerations. First of all, we present several implementations to simulate the effect of the TCIF on a digital representation of an image, and compare them with recorded images. Then, we offer some more insight into the difference between the TCIF and the classical CTF, and how this difference may affect interpretation of experimental data. Finally, we use the forward TCIF simulation to evaluate and optimize the performance of techniques that aim at correcting the effects of the imaging function and at recovering the original sample projection.

Background

In our first work on the tilted contrast imaging function (TCIF, Philippsen et al 2007B), we have laid the theoretical foundation for describing th imaging process for weak-phase objects that are tilted in regard to the lens system. We have shown that this imaging process cannot be described by a convolution, and hence removing the resulting optical artifacts from the images is more involving than

for non-tilted images, where the contrast transfer function (CTF) is valid.

The previously derived mathematical description of the TCIF is given by

$$Q(\mathbf{p}) = i\left\{e^{-iW_0(\mathbf{p})}\,\Phi\left(\mathbf{p} - \tfrac{1}{2}\mathbf{d}p^2\lambda\tan\alpha\right) - e^{iW_0(\mathbf{p})}\,\Phi\left(\mathbf{p} + \tfrac{1}{2}\mathbf{d}p^2\lambda\tan\alpha\right)\right\}. \tag{1}$$

Here, Q is the resulting measurement in Fourier space, Φ is the Fourier representation of the sample, **p** is the 2D frequency vector parallel to the image plane, **d** is a unit vector perpendicular to the tilt axis, and α is the tilt angle. The term W_0 is based on the Scherzer formula (Scherzer, 1949), and is given by

$$W_0(\mathbf{p}) = \frac{\pi}{2}C_s\lambda^3p^4 - \pi\lambda z_0p^2 + \frac{\pi}{2}z_a\lambda p^2\sin\left(2\chi - \chi_0\right), \tag{2}$$

where C_s is the spherical aberration of the microscope, λ is the electron wavelength at the microscope acceleration voltage, z_0 is the base defocus in the center of the sample, z_a is the defocus difference due to astigmatism, χ_0 is the reference angle used to describe the direction of the astigmatism[1], and $\chi = \tan^{-1}\frac{p_y}{p_x}$. For the case that the tilt angle is zero, Eq.(1) reduces to the well known CTF form

$$Q(\mathbf{p}) = 2\sin\left[W_0(\mathbf{p})\right]\Phi(\mathbf{p}) \tag{3}$$

[1] The original TCIF paper did not contain the term for the astigmatism, we have since augmented Eq(2) to include it as well.

Results and Discussion

Thon Rings in Experimental and Simulated Images

A set of carbon film images was collected at 0, 30, 45 and 60 degree tilt angle, and a matching set of simulations of carbon film was subjected to a forward TCIF transformation (Eq.(1)), with parameters taken from the experimental setup, as detailed in the Methods and Implementation section. Power spectra of experimental and simulated images were visually matched and compared, as shown in Figure 1. The envelope function that is traditionally used to represent the quenching of the signal power at high frequencies was purposefully left out of the simulations. At first sight, the Thon ring patterns show excellent agreement between the experimental and simulated power spectra. At low tilt, the visible Thon rings in the TCIF simulated images extend beyond those in the respective experimental tilt images. This is a consequence of the lack of the envelope function in the TCIF formulation, and is especially prominent at zero tilt (Figure 1A). At higher tilts, a fading of the visible Thon rings is evident in the experimental power spectra. The effect is very well matched by the simulated counterparts, especially for very high tilt angles (60 degree) (Figure 1D). Given the absence of the quenching envelope in the TCIF, this fading effect must have a different reason.

Thon Ring Profiles

To demonstrate the nature of this optical phenomenon, one-dimensional profiles of two tilt conditions were calculated, based as above on simulated carbon film. A 0 degree tilt profile was obtained by transformation of the simulated carbon film by the normal CTF as given in Eq.(3), but multiplied with an envelope function to take the effects of the spatial and temporal incoherence into account (Reimer, 1997). A 60 degree profile was obtained by a TCIF transformation on the simulated carbon film. Once again, the signal-quenching envelope function was purposefully left out of the model. Although both profiles feature disappearing Thon rings, they fade out in strikingly different ways (Figure 2).

For the untilted case, the envelope function in the CTF causes a quenching of the signal, which eventually drops below the noise level (Figure 2A); this is the well known effect of the beam incoherence, and is usually sufficient to account for the disappearance of the Thon rings in untilted images. The oscillations become less and less wide. This translates into a fading of the Thon rings (power spectrum in the inlay of Figure 2B), but also in a general drop of the signal (since our

simulations only include "sample" noise from the carbon film, without any electron statistical noise, the signal eventually fades to nothing).

The profile from the TCIF simulation (Figure 2B) exhibits a very different behaviour: the oscillations' amplitude decreases, but fades together into a signal clearly above the noise level, and without the CTF characteristic zero crossings. The Thon rings disappear in the power spectrum (inset of Figure 2B), but the general signal level does not drop; the seemingly present noise in the outer regions of the power spectrum still contains information coming from the underlying sample, the carbon film. This result comes from a numerical simulation; of course, in power spectra of real tilted experimental images, the quenching effects from the spatial and temporal incoherence are also present, and eventually drop the signal below the noise level at high frequencies. However, the TCIF induced fading seems to be the more prominent effect at higher tilt angles (Figures 1 and 2) and therefore our TCIF formulation can faithfully reproduce the Thon ring patterns at such conditions, even without an additional envelope function.

This analysis shows how two completely different causes (the instrument's incoherence and the tilt) can have similar effects (the disappearance of visible Thon rings) on the power spectra of electron microscopy images. Since the different nature of the two phenomena makes the interpretation and the processing of tilted and untilted images very different, the necessity to clearly recognize their presence and account for their separate effects cannot be overemphasized.

TCIF Delta Notation

In order to pinpoint the source of this TCIF specific fading, an alternative notation of Eq.(1) is used, which splits the TCIF into a CTF Sample Modulation Term and a Sample Dependent Delta Term. Using the following notation

$$\begin{aligned}\Phi\left(\mathbf{p}-\tfrac{1}{2}\mathbf{d}p^2\lambda\tan\alpha\right)&=\Phi\left(\mathbf{p}\right)+\Delta_1(\mathbf{p})\\ \Phi\left(\mathbf{p}+\tfrac{1}{2}\mathbf{d}p^2\lambda\tan\alpha\right)&=\Phi\left(\mathbf{p}\right)+\Delta_2(\mathbf{p})\end{aligned} \tag{4}$$

with $\Delta_1,\Delta_2\in\mathbb{C}$, Eq.(1) can now be rewritten as:

$$Q(\mathbf{p})=\underbrace{2\sin\left(W_0(\mathbf{p})\right)\Phi\left(\mathbf{p}\right)}_{\text{CTF Sample Modulation Term}}+\underbrace{i\left\{e^{-iW_0(\mathbf{p})}\Delta_1(\mathbf{p})-e^{iW_0(\mathbf{p})}\Delta_2(\mathbf{p})\right\}}_{\text{Sample Dependent Delta Term}} \tag{5}$$

While the behaviour of the CTF Sample Modulation Term is predictable, because it follows the one of

the modulating CTF function, the behaviour of the Sample Dependent Delta Term is strongly affected by the underlying sample values; therefore, multiple runs with randomized carbon film simulations have to be used to generate plots of this TCIF Delta notation. The detailed procedure is outline in the Methods and Implementation section, and the resulting amplitude profiles for 5, 30 and 60 degree tilt are shown in Figure 3.

The first term in Eq.(5) - the CTF Sample Modulation term - reproduces the well known Thon ring pattern, and thus the corresponding intensities in Figures 3A, 3B and 3C bear no surprise behaviour. Since it contains a multiplication between the CTF kernel and the Fourier representation of the original sample, the well known plotting of the real function $2\sin\left(W_0(\mathbf{p})\right)$ allows the sample independent modulation to be represented easily.

The second term in Eq.(5) - the Sample Dependent Delta Term - is not as straightforward. It is a linear combination of two different complex values of the Fourier representation of the sample, each with a frequency dependent phase shift. Therefore, it does not exhibit a sample independent pattern as the CTF Sample Modulation term. As seen in the corresponding amplitudes in Figures 3A, 3B and 3C, its relative contribution to the overall amplitude increases with increasing frequency, the higher the tilt, the faster the increase.

We find the explanation for the TCIF-specific Thon rings fading effect (Figures 1 and 2) in the relative contributions of these two terms: At low spatial frequencies, when the contribution of the Sample Dependent term is low, the characteristics of the CTF Sample Modulation Term, i.e. the Thon rings, stand out. At higher and higher frequencies, the Sample Dependent Delta Term contributions overlays the Thon rings from the CTF term, thus causing them to fade away, independent of any envelope.

It should be no surprise to see that the amplitude of the Sample Dependent Delta Term can reach values higher than the maximum of the CTF Sample Modulation Term. It can be easily derived mathematically that if Φ_{max} is the term with the maximum amplitude in the Fourier representation of the sample, the CTF Sample Modulation Term can have a maximum amplitude of $2\left|\Phi_{max}\right|$, while the Sample Dependent Delta Term can in principle reach an amplitude of $4\left|\Phi_{max}\right|$.

It is also important to note that due to the presence of the Sample Dependent Delta Term on top of the CTF Sample Modulation Term, the TCIF has no zero-crossings and thus - in theory - no information loss occurs during the imaging process under tilted conditions. It has been shown previously

(Philippsen et al., 2007B), that a full inversion of the TCIF model is theoretically possible, with information from the sample function fully recovered at all spatial frequencies. It has to be seen if experimental images, where the high noise level sometimes floods particularly weak signal, allow the recovery of all the information that is carried through the imaging process. Furthermore, the information at high frequencies is "scrambled" by the effect of the TCIF, and might not be straightforward to extract.

Including the Phase

Up to this point, we have limited ourselves by looking at amplitudes and power spectra only. This is perfectly alright when discussing non-tilted images transformed by the CTF, since the Fourier space modulation is caused by a real function, and the only effect on the phase is a flipping. In case of the TCIF, however, the second term in Eq.(5) already hints at a sample and tilt dependent modification of the phase that goes beyond simply flipping.

Using the same profile procedure based on the TCIF Delta form mentioned above, we can gain insight into the phase changes imposed by the TCIF on the Fourier representation of the sample. In Figure 4, absolute phase differences between the Fourier representation of the sample and its CTF as well as TCIF transformation are plotted on top of each other.

The effect of the CTF is the expected sharp step between a phase difference of 0 and a phase difference of 180 degrees, depending on the sign of the CTF kernel. The effect of the TCIF cannot be described by a simple phase flip. Instead, the average phase difference changes continuously, following oscillations of varying width and amplitude. The width in frequency of these oscillations and the pattern followed by their amplitudes strongly depend on the tilt geometry parameters, but are roughly delimited by the corresponding phase flips of the CTF.

Two important observations can be deduced from the analysis of these simulations (Figures 3 and 4). The first is that the phase behaviour of the TCIF model diverges from the CTF model, even at frequencies where the CTF term in Eq.(5) is still dominant and the Thon rings are still visible in the power spectrum. This implies that when phases are concerned, the CTF model does not approximate the imaging process in tilted geometry, even at frequencies where the amplitude of the Sample Dependent term is low, and the Power spectra would lead one to believe that the CTF is still a valid description.

The second observation is that a correction of the TCIF effect is required to recover accurate phase

information. The alternatives, performing no correction at all or correcting tilted images according to the CTF model, introduce phase errors: In the first case, the error would corresponds to the difference between the TCIF phase difference line (blue) in the plots of Figure 4 and the zero line. In the second case, the error would be equivalent to the difference between the CTF phase difference line (red) and the TCIF phase difference line (blue).

Correction Schemes under Scrutiny

In our previous Work (Philippsen et al., 2007B), we presented an inversion strategy that allows the recovery of the original sample function from the observed image, assuming that the parameters of the TCIF model are known. Due to the absence of zero crossings in the TCIF model, a complete recovery of the sample information could be achieved. This technique is however extremely demanding from a computational point of view, requiring the solution of systems of linear equations with thousands of unknowns. Although the validity of the approach has been rigorously tested on small artificial images, its application to images of realistic size appears to require more investment into efficient numerical methods.

Since the problem of reversing the effects of the imaging function in tilted samples is not new, several indirect correction strategies have been developed over the years (Winkler et al. 2003,Fernandez et al., 2006), some applicable only to specific electron microscopy techniques, other of general nature. The main advantage of having algorithms that implement the TCIF model is the ability to generate simulated tilted images for which the original sample function is known. Programs and algorithms that correct the effects of the imaging process can then be directly applied to the simulations, and subsequently the corrected images can be compared to the originals. The performance of different correction strategies can be evaluated and discussed, and its dependency on the input parameters can be analyzed.

The most generic class of correction schemes exploits the fact that the defocus is constant in image stripes parallel to the tilt axis. Within each stripe the TCIF can be approximated with good accuracy by the classical CTF. Established Wiener-filter based deconvolution techniques can be used for the correction (Grigorieff, 1998).

Several implementations of this concept exist. We chose to study and compare two: one developed by Fernandez et al. (2006) for electron tomography (ET), and the other developed in our group and based on the principles of linear optics (See Material and Methods section). Our analysis focused on the

recovery of the original sample function phase information. To allow a more direct comparison, the custom deconvolution filter employed in Fernandez's method was substituted with a standard Wiener one. Initially the dependence of the correction performance on the width of image stripes used by the two methods was evaluated. In a second step, the accuracy of the phase determination using both strategies was compared to the correction a tilted image using a traditional untilted CTF, and with the effect of performing no correction at all.

The TCIF model parameters for the simulations were chosen in a way that reproduced typical ET experimental conditions. Specifically, two experimental setups currently used in ongoing tomography research projects were simulated (M. Eibauer, MPI Martinsried, and B. Zuber, MRC Cambridge, personal communications). The main differences between the two setups lie in the image pixels (0.661nm for the first and 0.374nm for the second) and in the defocus distance used when images were collected. (2μm for the first and 4 μm for the second). The value of the Wiener filter constant for the stripe correction methods was chosen as 10% of the average signal, a typical value used in image processing of micrographs (Grigorieff, 1998). A detailed description of the other parameters can be found in the Methods and Implementation section.

Starting from random sample functions, simulated images of samples tilted at 60 degrees, a typical high tilt angle in Electron Tomography, were generated. The correction schemes were then applied to the simulations and the average recovered phase was compared to the original at all spatial frequencies.

The average absolute phase recovery error at different spatial resolutions and for different stripe widths is shown in Figure 5. Plots A and C were generated using the correction method developed by Fernandez. The performance of the linear optics scheme is shown in plots B and D. Results obtained with different widths of the stripe images are shown on the plot using different colors (see figure legend).

Under the first experimental setup (Figure 5, plots A and B), the performance of the two approaches is comparable at high resolution, with the Fernandez method showing a weaker dependence on the stripe width. This is expected: A broader strip results in a coarser approximation of the local defocus for the linear optics method, where a whole area of the image is processed only once using the defocus at the centre of the strip. In the Fernandez method the processing of the image stripes is shifted by just one pixel at each step, allowing for a fine assessment of the local defocus. It should also be noted that using too narrow image stripes compromises high resolution performance with both methods. This happens

when the image stripe is narrower than the point spread function at that particular defocus: information gets cut during the correction process. When moving from a high to an intermediate resolution, the error in phase recovery for both correction schemes decreases markedly. This happens abruptly and appears as a “step” in the plots. The phase error stays then uniformly low over a wide range of spatial resolutions. At resolutions of 100 Å and below, however, the phase error for the Fernandez method appears to increase and becomes larger and larger as the resolution decreases. This error does not appear to be caused by the correction strategy itself. The Fernandez algorithm cannot correct the whole image. Stripes parallel to the tilt axis and close to the edges of the micrograph are excluded from the correction. This introduces an edge effect at the stage where the phase of the corrected image is compared with the phase of the original sample function. In turn, this causes the apparent error increase at low resolution. As will be dicussed later (Figure 6), a similar effect happens also when correcting the image using a traditional CTF function, and even when performing no correction at all. These three procedures all have the edge effect in common.

Similar simulations were run for the second experimental setting (Figure 5, plots C and D). As stated above, the main differences between this setting and the previous one are the pixel size and the defocus value. Using a higher defocus makes the radius of the spread function for a particular pixel wider, and a small pixel size makes it easier to lose information when the stripe width is too small. The performance of the linear optics method appears to be slightly less sensitive to change in the stripe width at extremely high resolution, but the general trend is clearly similar to the previous case. It is important to point out that the “step” in the error plots moved to a lower resolution. Additional tests were also carried out using different values of the Wiener filter constant (1% and 0.1% of the average signal strength). While the results were slightly different, the general trend showed the same behaviour, thus they are not shown in this manuscript.

In the second phase of the simulation experiment, three different correction strategies were applied to the same simulated tilted image (a stripe based correction, a correction using a classical untilted CTF no correction at all). The phase errors of the three methods were then drawn on a single plot, allowing an easy comparison. In plots A and C in Figure 6 the Fernandez method was used for the stripe correction. In the plots B and D the Linear Optics method was used. Plots A and B refer to the first setup, while the plots C and D at the bottom show data from the second one.

All plots show an abrupt “step” in the phase error at the border between intermediate and high resolution. It is clear from the plots that this happens when the oscillations of the imaging function start

to have an impact on the phase recovery. This makes the exact resolution where this critical point lies dependent on the defocus used to collect the image. At resolutions lower than this point, correction of the effects of the TCIF brings no clear advantage. On the contrary, performing no correction at all seems to be the choice that minimizes the error. This information has been known intuitively by many researchers in the field for years. However, as far as we are aware, this is the first time that it has been quantified using rigorous imaging simulations. At resolutions where the fluctuations of the imaging function have an impact on the phase recovery process, the plots clearly show some form of correction to be mandatory; the phase error for uncorrected images becomes too large. Heuristically correcting a tilted image using a classical untilted CTF function and the defocus value at the center of the image, although conceptually not correct, already gives an improvement. However, performing a full stripe-based correction shows a drastic reduction of the phase error. Both correction methods analyzed in this manuscript have proven to be equally efficient at improving the accuracy of the phase recovery.

When the goal of structural studies is intermediate resolution (in the experimental settings investigated in this work, on the order of 40 $Å^{-1}$ or worse), correcting the effects of the TCIF function does not seem to be worth the effort. However, when high resolution information needs to be extracted from the data, two possible strategies are available. The first is lowering the defocus. This makes the oscillations of the imaging function start at higher spatial frequency and pushes the threshold of where correction need to be applied to higher resolution. However, this also causes a loss of contrast that becomes weaker and weaker as the defocus is lowered. The alternative is keeping a relatively high defocus but performing a correction using one of the stripe correction methods discussed in this manuscript.

A great variety of experimental settings and correction schemes can be replicated in silico, in addition to the ones presented in this work. Other properties can influence the choice between different methods, like speed and computational requirements. These concerns are beyond the scope of this manuscript. It is however clear that the availability of the TCIF model and the possibility to generate simulations of tilted images allows a deeper understanding of the imaging process, a rational planning of the image processing for data collected on tilted specimens, and the fine tuning of all correction parameters. This type of analysis can be conducted during short, convenient simulation runs before the start of time-consuming full correction procedures.

Methods and Implementations

All described algorithms and simulations were implemented, run and analyzed with IPLT (Philippsen et al, 2007A), unless otherwise noted.

CTF and TCIF Profiles

To obtain radial profiles of the optical effects under tilted and untilted conditions as exhibited in Figure 2, two simulations generating 1D images with a length of 1024 pixels were run. The first produced images tilted at an angle of 60 degrees (along the length of the image) using the standard TCIF model. The second generated untilted images using the classical CTF model modulated by an envelope function to account for the effects of the spatial and temporal incoherence of the instrument's emissions (Reimer, 1997). Identical parameters where used for each image (as detailed in the figure legend). For each simulation, the power spectra from all the runs were averaged and a profile of the result was finally computed.

TCIF Delta Form Profiles

Based on the alternative TCIF Delta formulation in Eq.(5), two TCIF model simulations were run generating 1D images with a length of 1024 pixels. The tilt was chosen to be of 30 degrees for the first simulation and of 60 degrees for the second. The defocus at center of all images was set to 2μm and the pixel sampling to 2 Å. The other parameters for the TCIF model were: spherical aberration 2mm, acceleration voltage 200kV. Each simulation performed 1000 runs applying the TCIF model to different random film simulation images. For each run, the values of the CTF Sample Modulation term and of the Sample Dependent Delta Term for all spatial frequencies were computed.

Ampltitude and phases from all runs were averaged for each term. In Figure 3, showing amplitude profiles, areas below the plot line were filled with color in order to emphasize the relative weight of the two terms. In Figure 4, representing phase profiles, only the plot line is colored.

Figure 4 also features a CTF phase profile. This was obtained by running a single simulation run using the parameters described in the previous paragraph, but a 0 degrees tilt angle.

Stripe-based correction scheme: Linear Optic Protocol

This algorithm relies on the fact that the imaging process follows, in first approximation, the physical principles of linear optics. Each stripe is then considered in the context of the whole image system, and the result of its processing is linearly summed to the results of the processing of the other columns. The algorithm can be briefly described using the following sequence of operations:

1. A binary mask is created that selects from the original uncorrected image a stripe parallel to the tilt axis. An integer divisor of the image width is chosen as width of the stripe.
2. CTF correction using a wiener filter is performed on the masked image. The defocus value used for the correction is the defocus at the center of the stripe, computed using basic trigonometry considerations.
3. A new mask which selects the adjacent non-overlapping stripe is created and applied to the input image. Steps 2 and 3 are then repeated until the whole input image is processed.
4. All the corrected images are summed into the final output image.

Analysis of performance of correction schemes

Two specific experimental setups were simulated. In the first (Eibauer, personal communication), 1D images with a length of 2000 pixels and with a pixel sampling of 0.661 nm were created. The defocus at the centre of the images was set at 2μm. All simulations had a tilt along the length of the image of 60 degrees. The acceleration voltage was chosen to be 200 kV and the spherical aberration constant 2 mm. In the second setup (Zuber), images with the same size and tilt geometry were simulated. However, a pixel sampling of 0.374 nm, a spherical aberration constant of 2.2 mm and an acceleration voltage of 300 kV were used. Several correction strategies were applied to the simulated tilted images: a traditional Wiener filter-based CTF Correction (Grigorieff, 1998) using the defocus value at the centre of the image, the stripe-based correction scheme developed by Fernandez (Fernandez et al., 2006) (reimplemented in IPLT) and the linear optics correction scheme described in the previous section. Each stripe-based correction scheme was applied using 4 different stripe widths. For the first experimental setting, the following stripe widths were used: 264nm (400 pixels),132nm (200 pixels), 66nm (100 pixels), and 17nm (25 pixels). In the second experimental setting, correction was performed with strip sizes of 150 nm (400 pixels), 75nm (200 pixels), 37 nm (100 pixels) and 9nm (25 pixels). All simulations performed 50 runs. For each run the corrected image was compared to the input image

to which the simulated imaging process was applied. The uncorrected image was also compared to the input image. Phase differences for each pixel in the Fourier Transforms of the images were computed. Phase differences for the same pixels were averaged across all runs. The method developed by Fernandez leaves areas near the edges of the image, perpendicular to the tilt axis, where correction cannot be performed. This effect was taken into account and the areas were not considered during the comparison.

Experimental Images

A series of images of carbon film were collected on a Philips CM 200 FEG microscope equipped with a Gatan Ultrascan 1000 CCD camera (2048x2048 pixels). The series included images collected at tilt angles of 0, 30, 45 and 60 degrees at a magnification of 115000x, resulting in a pixel size of 1.217 Å. For image processing the images were shrunk to a size of 1024x1024. The defocus for each image was measured by manually fitting a CTF model to the power spectrum, using the IPLT CTF overlay. Defocus values for the tilt series were found to be: 5.55 μm (untilted image), 6.77 μm (30 degrees), 5.35 μm (45 degrees) and 5.65 μm (60 degrees).

Film Simulation

Simulated EM images of carbon film were generated using a method proposed by Boothroyd (Boothroyd, 2000) and based on previous work by Chevalier and Hytch aimed at simulating images of generic amorphous materials (Chevalier and Hytch, 1993). This approach is briefly summarized in the following paragraphs, and has been implemented in IPLT.

The carbon film is seen as a sequence of slices, each 1nm thick, laid one on the top of another. The number of slices depends on the simulated film thickness. The distribution of atoms within a slice is considered random, and the exact location of each atom is ignored. The potential projected by each slice is written as $V(\mathbf{r}) = \rho(\mathbf{r})v(\mathbf{r})$, where **r** is a planar coordinate vector in real space, ρ is the atomic density at the coordinates defined by **r,** and v(**r**) is the scattering potential at the same coordinates. The projected density is represented using Poisson-distributed noise. The λ parameter of the Poisson distribution is proportional to the average atomic density in the slice. The noise is weighted in Fourier space with the real scattering factor of amorphous carbon, which is considered to be equivalent to the one of a single atom of carbon. The value of the scattering factor at an arbitrary scattering angle is interpolated from the table provided in the International Tables for Crystallography (First online

edition, 2006). The density projected by all the slices is summed up in the final image. The increased film thickness that the electron beam traverses in tilted geometry is simulated by increasing the number of slices according to simple trigonometric considerations.

TCIF Simulation

Explicit Fourier Summation

As discussed in the first TCIF paper, the calculation of $\Phi(\mathbf{p})$ in Eq.(1) at arbitrary frequency values based on a discrete real-space representation $\phi(x_n, y_n)$ with sampling Δ is done using an explicit Fourier summation:

$$\Phi\left(p_x, p_y\right) = \sum_{b=0}^{N_b-1} \sum_{a=0}^{N_a-1} \phi\left(x_a, y_b\right) e^{-2\pi i(p_x a\Delta_x + p_y b\Delta_y)} \tag{6}$$

Combining Eq.(1) and Eq.(5), we obtain:

$$\begin{aligned} Q\left(p_x, p_y\right) &= 2\sum_{b=0}^{N_b-1} \sum_{a=0}^{N_a-1} e^{-2\pi i(p_x a\Delta_x + p_y b\Delta_y)} H\left(p_x, p_y, x_a, y_b\right) \phi\left(x_a, y_b\right) \\ H\left(p_x, p_y, x_a, y_b\right) &= \sin\left[W_0\left(p_x, p_y\right) - \pi(p_x^2 + p_y^2)\lambda(a\Delta_x \cos\beta + b\Delta_y \sin\beta)\tan\alpha\right] \end{aligned} \tag{7}$$

This formulation allows a precise but slow implementation of the TCIF simulation. Two additional algorithms have been designed to produce faster simulations, given below.

Tilt Axis Rotation Algorithm

For a discretized dataset (i.e. a digital image), assuming that the tilt axis is parallel to the y axis of the reference system, Eq.(7) can be reduced to a 1D transformation:

$$Q_{p_y}\left(p_x\right) = 2\sum_{n=0}^{N_x-1} e^{-2\pi i p_x n\Delta_x} \tilde{H}_{P_x}\left(p_x, n\right) \tilde{\Phi}_{p_y}\left(x_n\right) \tag{8}$$

Δ_x stands for the pixel sampling width in the x direction . $\tilde{\Phi}_{p_y}$ is the Fourier Transform of the dataset in the y direction:

$$\tilde{\Phi}_{p_y}\left(x_n\right) = FT_y\left[\phi(x_n, y_n)\right] \tag{9}$$

$\tilde{H}_{P_x}$ is:

$$\tilde{H}_{P_x}\left(p_x, n\right) = \sin\left[W_0\left(p\right) - \pi p^2 \lambda n \Delta_x \tan\alpha\right] \tag{10}$$

In cases where the tilt axis is not parallel to the y axis, the input image is rotated, the transformation is applied, and then the image is be rotated back to its original orientation.

Shannon Window Interpolation Algorithm

An approximate value for the Fourier Transform of ϕ at an arbitrary continuous frequency **p** can be computed from a small set of discrete samples of the Fourier Transform itself using the Moving Window Shannon Reconstruction (MWSR) technique (Lanzavecchia and Bellon, 1997). The Fourier Transform of the input image is seen as a discrete sampling of the Fourier Transform of ϕ. The value of at frequency **p** is interpolated from a 2D window of discrete frequencies surrounding **p**. An appropriate interpolation kernel is used:

$$\Phi\left(\mathbf{p}\right) = \frac{1}{n_x n_y} \sum_{x=x_0}^{x_1} \sum_{y=y_0}^{y_1} \Phi\left(x, y\right) \frac{\sin\left(\pi\left(p_x - x\right)\right)}{\sin\left(\frac{\pi}{n_x}\left(p_x - x\right)\right)} \frac{\sin\left(\pi\left(p_y - y\right)\right)}{\sin\left(\frac{\pi}{n_y}\left(p_y - y\right)\right)} \cos^{A_x}\left(\frac{\pi}{n_x}\left(p_x - y\right)\right) \cos^{A_y}\left(\frac{\pi}{n_y}\left(p_y - y\right)\right) \tag{11}$$

Where x_0 and x_1 are the start and end frequency of the 2D window in the x direction, y_0 and y_1 are the start and end in the y direction, n_x is the number of samples in the x direction, n_y is the number of samples in the y direction, and A_x and A_y are interpolation parameters that satisfy the following condition: if the number of samples in one direction is odd, the corresponding A parameter must be even, and vice versa.

Using Eq.(11) to compute the Φ terms, the value of $Q(\mathbf{p})$ can easily be obtained from Eq.(1)

References

International tables for crystallography. First online edition (2006)

Boothroyd C.B., 1998. *Why don't high-resolution simulations and images match?* J Microscopy 190 pt 1/2, 99-108

Boothroyd C.B., 2000. *Quantification of high-resolution electron microscope images of amorphous carbon.* Ultramicroscopy, 83, 159-168

Chevalier J.-P., Hytch M.J, 1993. *Simulating the exit wave function for uniformly disordered systems.* Ultramicroscopy 52, 253-259

Grigorieff N., 1998. *Three-dimensional structure of bovine NADH: Ubiquinone Oxidoreductase (Complex I) at 22Å in ice.* J. Mol. Biol. 277, 1033-1046

Lanzavecchia S. and Bellon P.L. (1997). *The Moving Window Shannon Reconstruction (MWSR) in real and Fourier domain and its use in tomography***.** Scanning Microscopy Vol. 11, 155-169

Philippsen A., Schenk A.D., Signorell G., Mariani V., Berneche S. and Engel A., 2007. *Collaborative EM image processing with the IPLT image processing library and toolbox***.** Jou. Struct. Biolog. 157(1): 28-37.

Philippsen A., Engel H.A., Engel A., 2007. *The contrast imaging function for tilted specimens***.** Ultramicroscopy, 107(2-3) 202-212

Reimer L., 1997, *Transmission electron microscopy***.** Springer

Scherzer O., 1949, *The theoretical resolution limit of the electron microscope.* J. Appl. Physics 20

Winkler H., Taylor K.A., 2003, *Focus gradient correction applied to tilt series image data used in electron tomogrpahy.* J. Struct. Biol 143 24-32

FIGURE 1: Comparison of experimental (left) and simulated (right) power spectra of images of a carbon film in untilted geometry (A) and for tilt angles of 30 (B), 45 (C) and 60 (D) degrees. At low tilt angles the number of Thon rings that can be detected in the experimental images is much lower than in the simulations, which are not modulated by any envelope function. However, at high tilt angles the simulations predict the number of detectable Thon rings with remarkable accuracy, showing their disappearance to be caused by the TCIF, and not by the signal being lowered into the noise level. The images in this tilt series were collected at a magnification of 115000x and at the following undefocus distances: 5.55 μm (A), 6.77 μm (B), 5.35 μm (C) and 5.65 μm (D). The acceleration voltage of the instrument was 200 keV and its spherical constant 2 mm. The original size of the images was 2048x2048 pixels, but only the central 768x768 pixels of the power spectra are shown in the figure. The original pixel sampling was 1.217 Å. The dotted line corresponds to a spatial frequency in Fourier space of 10 A-1.

FIGURE 2 Comparison between the radial profile of the classical CTF (bottom) and the radial profile of the TCIF (top), based on simulated images under identical instrument conditions (see methods and implementations for more detail). Typical power spectra for the two cases are shown in the insets. While both show disappearance of the Thon rings at high frequencies, the radial profiles reveal this characteristic to be caused by different physical phenomena: The disappearance in the classical CTF function is caused by an envelope function (from the chromatic aberration and energy spread) that lowers the signal into the noise level. In the TCIF case, however, the disappearance of the Thon rings is an implicit consequence of the imaging function itself, and does not coincide with the fading of the signal into the background level. It can also be seen that there are no zero-crossings in the TCIF, as observed for the CTF, and thus - in principle - no information loss occurs. Simulation parameters for both conditions: Defocus (at centre) 5.65 um, spherical aberration 2mm, chromatic aberration 2mm (CTF), energy spread 2eV (CTF), aperture angle = 0.1 um, acceleration voltage = 200kV (lambda of 2.1pm)

FIGURE 3 Relative weight of the CTF Modulation Term and the Sample Dependent Delta Term in the power spectra of images of tilted samples. (See Eq. (5) and the Methods and Implementation section of this manuscript for a definition of the two terms). Three cases are shown: very low tilt (A, 5 degrees), average tilt (B, 30 degrees), and high tilt (C, 60 degrees). The top part of each figure shows the relative intensities of the two terms in the full power spectrum as a function of the spatial frequency. Values from 1000 random simulation form the scatter-plot. In the bottom part of each figure a typical power

spectrum is shown. At low tilt angles the contribution of the Sample Dependent Delta Term is weak at low spatial frequencies, but it increases steadily as the spatial frequency gets higher. For moderate and high tilt angles, the increase is much sharper and the contributions of the two terms reach the same weight at much lower frequencies. When this happens, the features of the CTF Sample Modulation Term, particularly the Thon rings, become less and less prominent in the power spectum, eventually fading out completely. The number of detectable Thon rings is then reduced. Important parameters used for the simulations are: defocus at the center of the image 2μm, image size 1024x1, pixel sampling 2 Å, spherical aberration 2mm, acceleration voltage 200kV.

FIGURE 4 Phase effects induced by the tilt geometry. A low tilt angle case can be seen on the left (30 degrees), while a high tilt case is shown on the right (60 degrees). The top part of the figure shows the phase difference between the sample function and the image observed after the simulation of the imaging process. The red line shows the phase difference for the classical CTF, while the blue line shows the same for the TCIF. At the bottom of the images, typical power spectra for the two cases are shown. Since the classical CTF only induces a flip of the phase of the input function, the phase error can only assume the value of 0 (no flip) or 180 (phase flipped). However, the TCIF induces more complex phase modifications. This happens even at low spatial frequencies where the features of the CTF Sample Modulation term of the TCIF are still predominant (See Figure 2) and where Thon rings can still be detected. Important parameters used for the simulations are: defocus at the center of the image 2μm, image size 1024x1024, pixel sampling 2 Å, spherical aberration 2mm, acceleration voltage 200kV.

FIGURE 5 Comparison of performance of two stripe-based correction schemes using different column widths. The figure show phase errors obtained by applying two different stripe-based correction schemes to simulated 60-degree tilted micrographs. On the x-axis, resolution (in Angstrom) is shown. On the y axis, the average phase recovery error is reported. All averages were computed over 50 random runs on images 2000 pixels wide. In plots A and C the correction was performed with the method published by Fernandez et al. (2006). The custom filter developed by Fernendez was however replaced with a standard Wiener filter for a more direct comparison. Plots B and D show the results obtained using a linear optics correction scheme (See Methods and Implementation section for details). Although data from 400Å to 10Å are shown in the plots, the resolution range that is most relevant for realistic electron tomography experiments (200Å to 20Å) has been emphasized using vertical dotted lines. Parts A and B of the figure shows the results obtained for a typical cell electron tomography

setup (Pixel size: 0.661 nm, defocus 2 μm). The different stripe widths are represented using colors: grey is a width of 264nm (400 pixels), red of 132nm (200 pixels), green of 66nm (100 pixels), and blue of 17nm (25 pixels). Parts C and D of the figure shows results for a typical experimental cryo-electron tomography setting (Pixel size: 0.374 nm, defocus 4 μm). Grey is a stripe width of 150 nm (400 pixels), red of 75nm (200 pixels), green of 37 nm (100 pixels), and blue of 9nm (25 pixels). This figure shows that both methods are relatively insensitive to the choice of stripe width, except for the extreme cases, which influence the performance at very high resolution. The approach developed by Fernandez displays an apparent weak performance at low resolution for all stripe widths. This is not caused by the correction procedure itself but by an edge problem introduced by the method which interferes with the error estimation algorithm (See the Results and Discussion section of the manuscript).

FIGURE 6 Comparison of average errors in phase recovery for different TCIF correction strategies: no correction at all (grey line), correction using an untilted CTF (red line), correction using a stripe-based approach (red line). Plots A and C refer to the stripe correction developed by Fernandez (Fernendez et al., 2006), plots B and D show results obtained using a linear-optics stripe approach (See Materials and Methods section). Typical cryo-tomography experimental conditions (Defocus 4 μm, Pixel sampling: 3.74 Å) were used for the simulations. The stripe width used for both methods was 37nm (100 pixels). The figure clearly shows that for studies which aim at medium resolution (up to about 4 nm under these specific conditions), correction of the TCIF effects is not required. However when the goal of the structural study is the extraction of high resolution information, some form of correction is mandatory. The two strategies analyzed in this manuscript have proven to be equally effective in performing the correction task. Data from 400Å to 10Å are shown in the plots, but the resolution range that is within reach of realistic electron tomography experiments (200Å to 20Å) has been bracketed within vertical dotted lines. All data in the plots are average from 50 simulation runs using different random film simulations. For an explanation of the high phase error at low resolutions in the experiments using the Fernandez methods, see the Results and Discussion section of the manuscript.

FIGURE 1

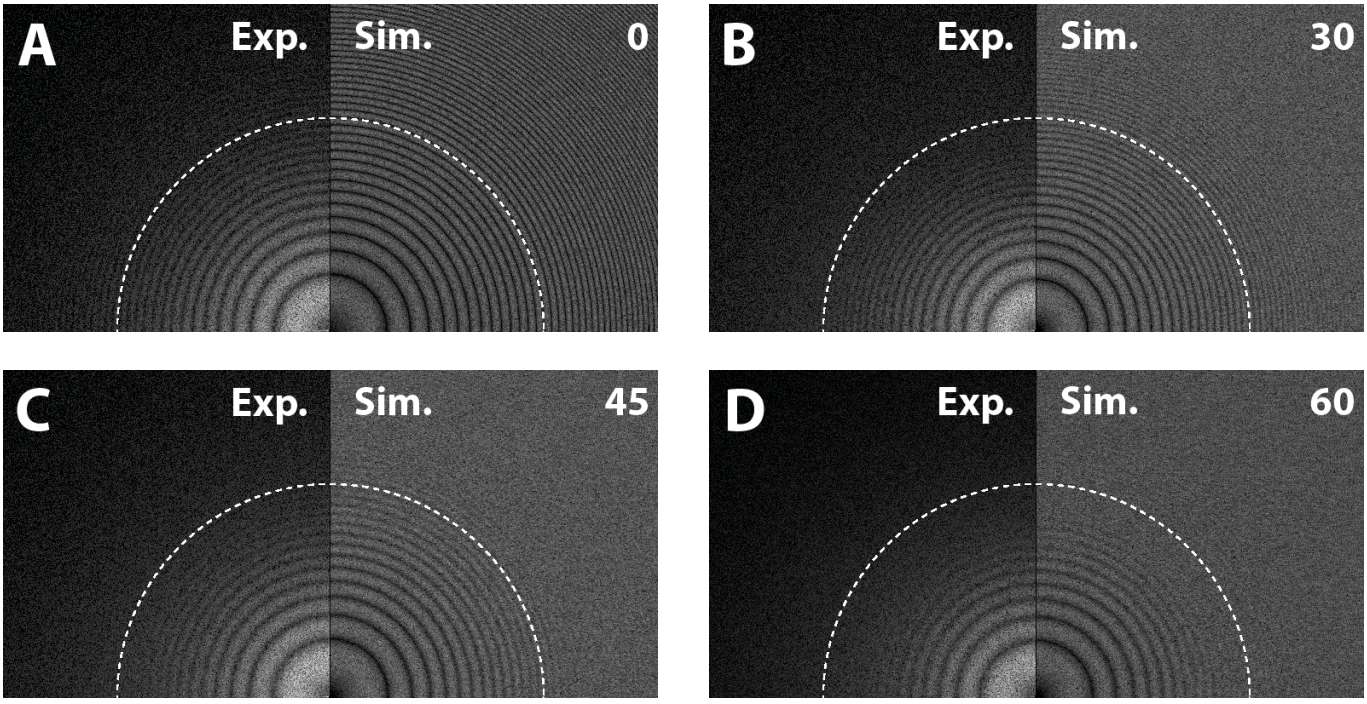


FIGURE 2

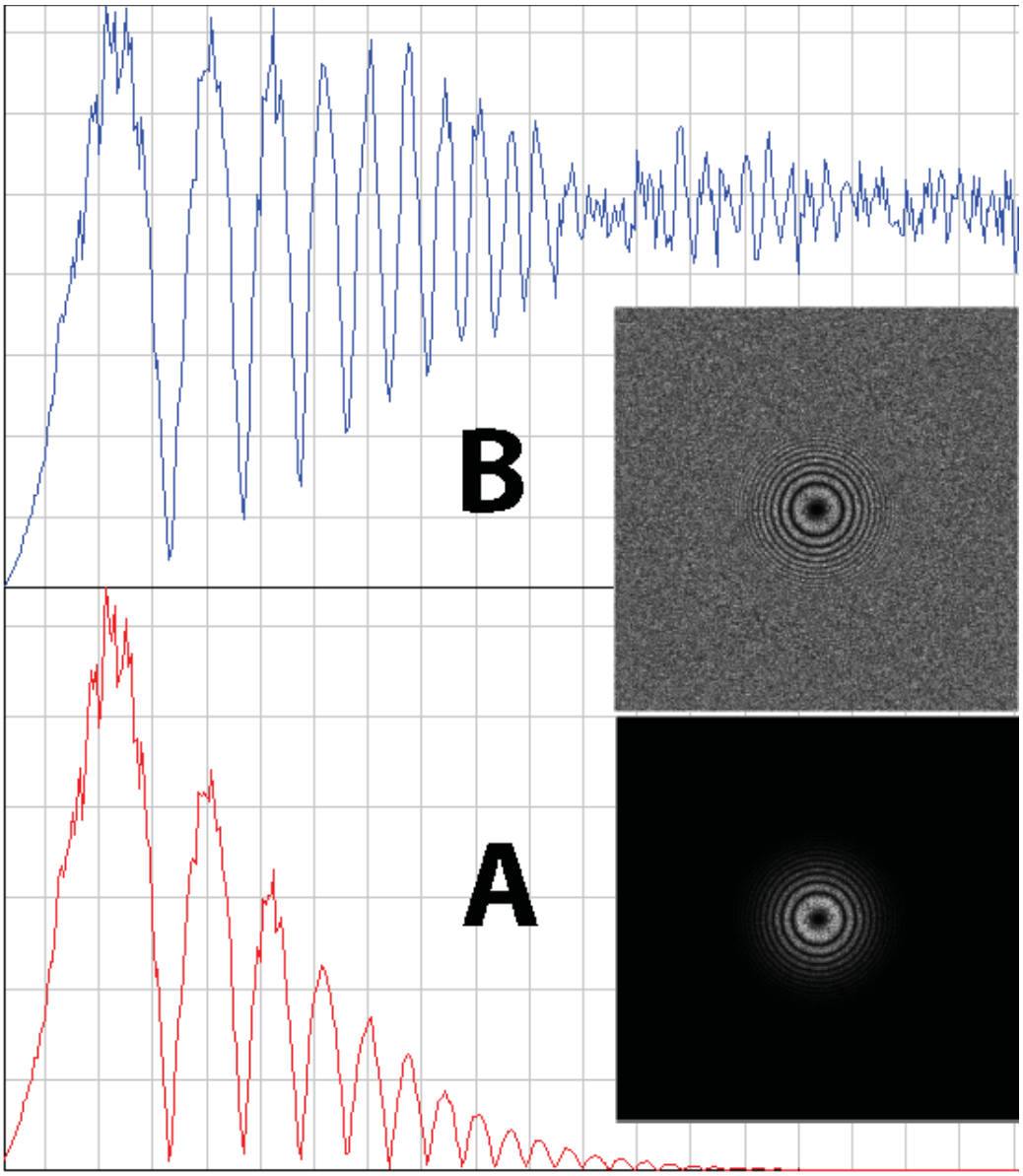


FIGURE 3

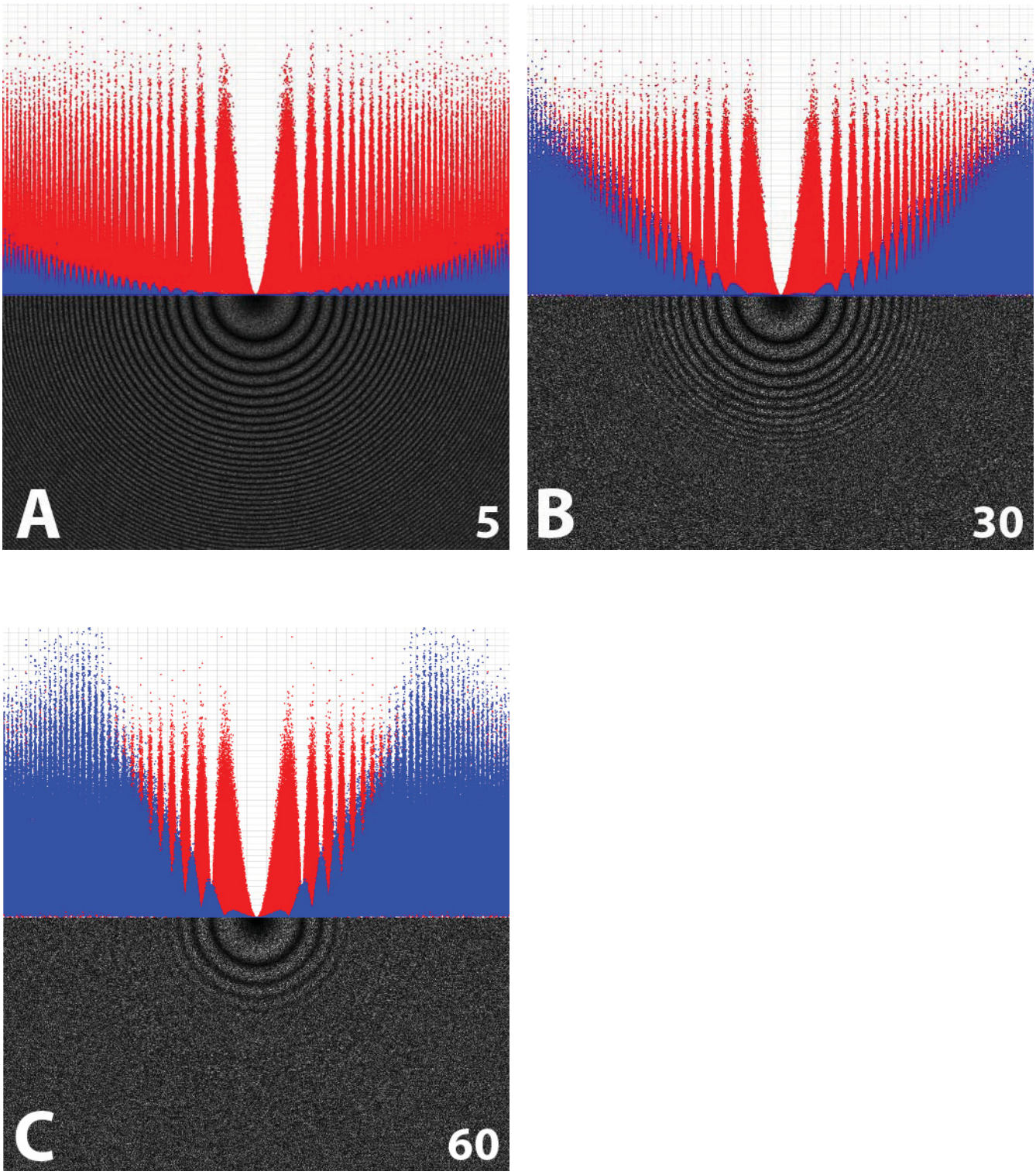


FIGURE 4

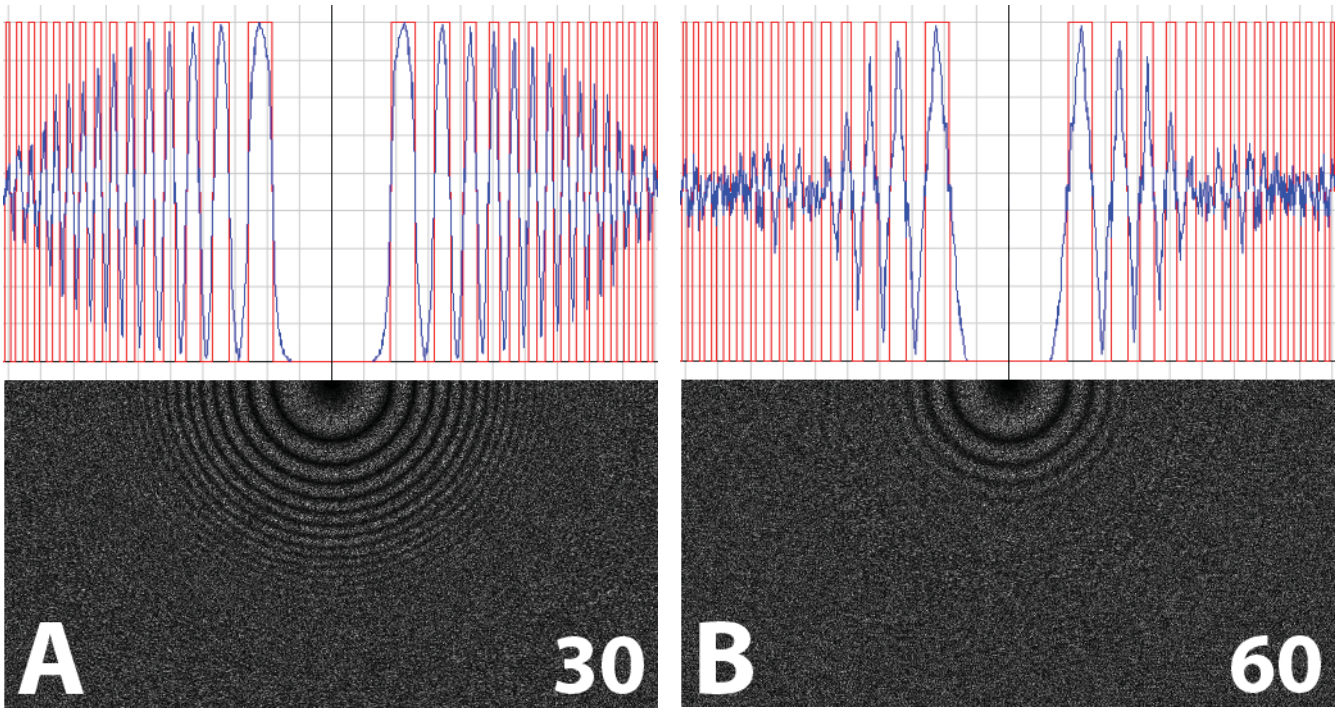


FIGURE 5

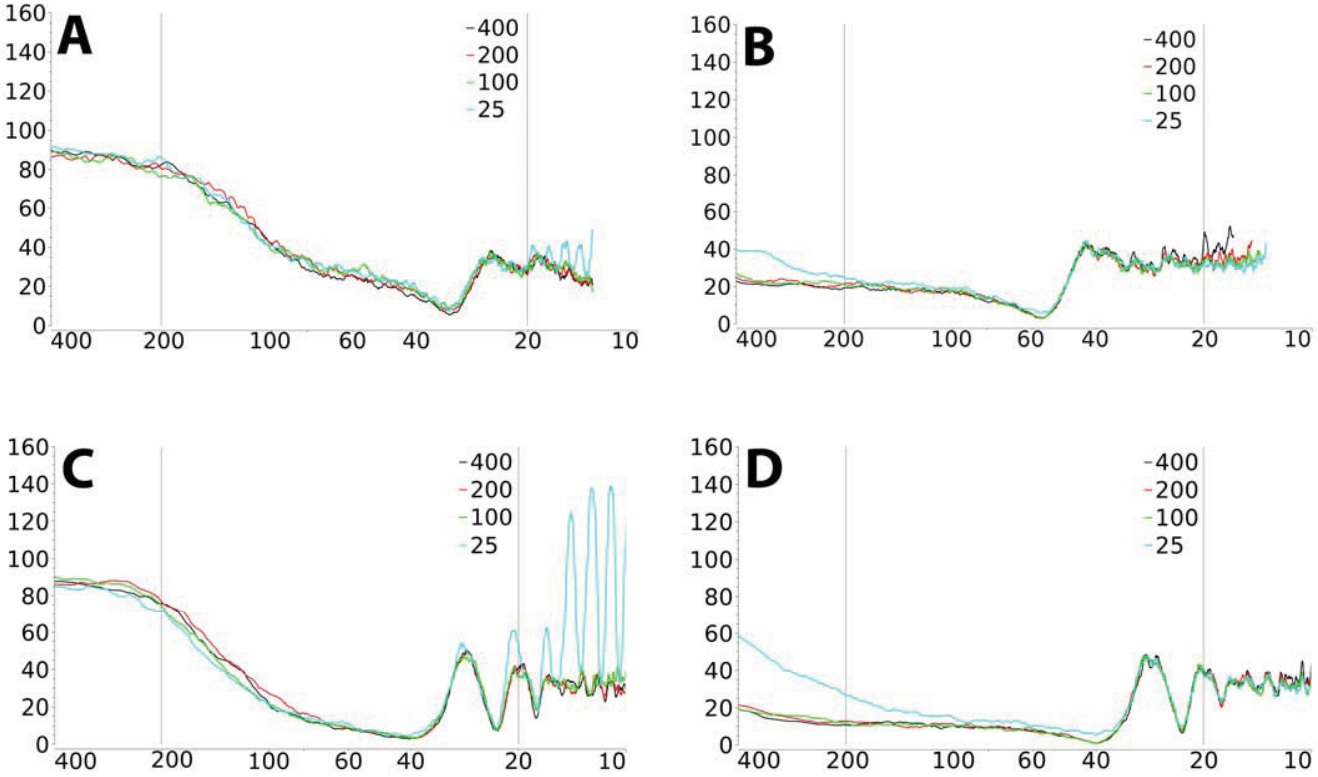
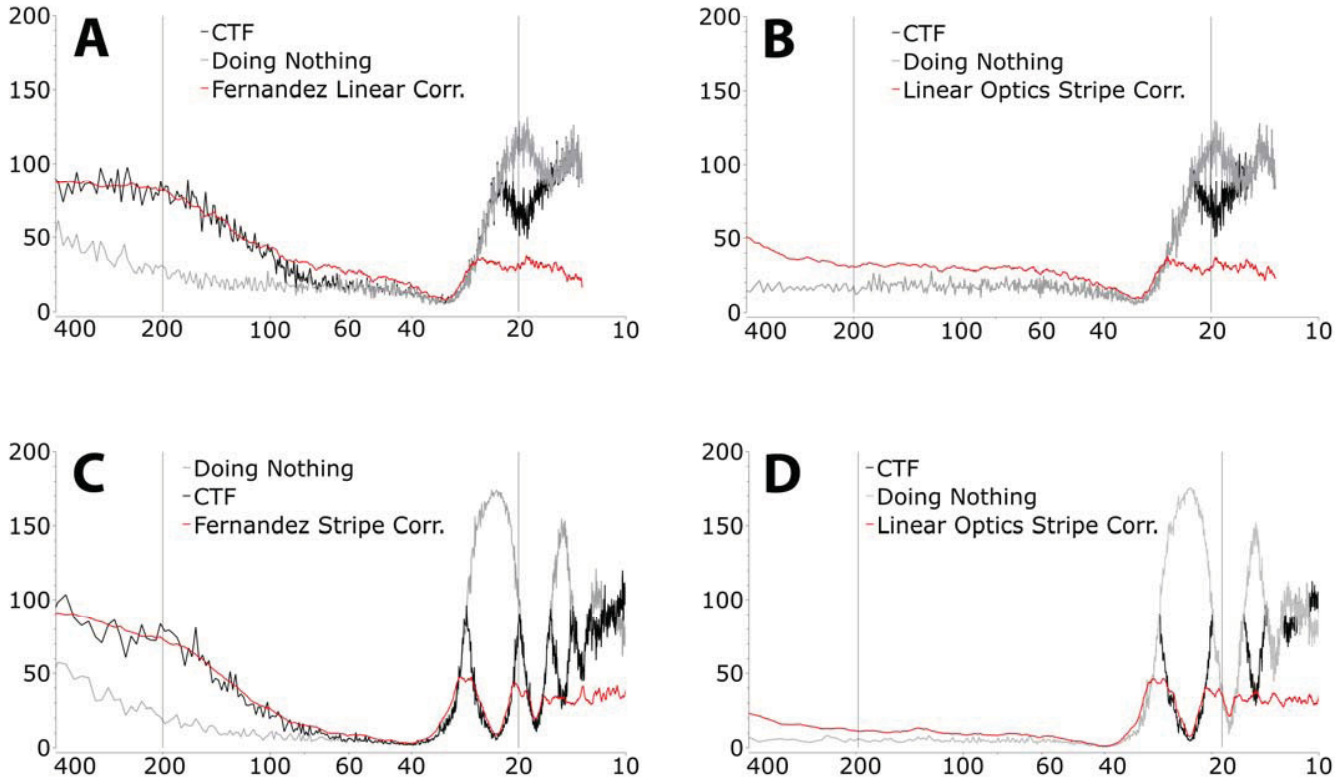

A
B
C
D
400
200
100
25
160
140
120
100
80
60
40
20
0
400
200
100
60
40
20
10

FIGURE 6


A
CTF
Doing Nothing
Fernandez Linear Corr.
B
CTF
Doing Nothing
Linear Optics Stripe Corr.
C
Doing Nothing
CTF
Fernandez Stripe Corr.
D
CTF
Doing Nothing
Linear Optics Stripe Corr.
200
150
100
50
0
400
200
100
60
40
20
10

Supplementary Material

Comparison of the various TCIF algorithm implementations

In order to compare the accuracy and speed of the various TCIF algorithm implementations, images of size 64x64, 128x128, 256x256 and 512x512 were generated using all available algorithms and the time necessary to obtain a complete simulation was measured. The 512x512 images generated using the different algorithms were then compared with each other. The image obtained using the Explicit Fourier Summation algorithm was assumed to be mathematically precise and taken as gold standard. The computational weight of this algorithm makes its use questionable on realistic-sized images. The comparison results clearly show a decrease in accuracy when switching from the Shannon Window Interpolation method to the Tilt Axis Rotation one, which is compensated by a remarkable increase in speed.

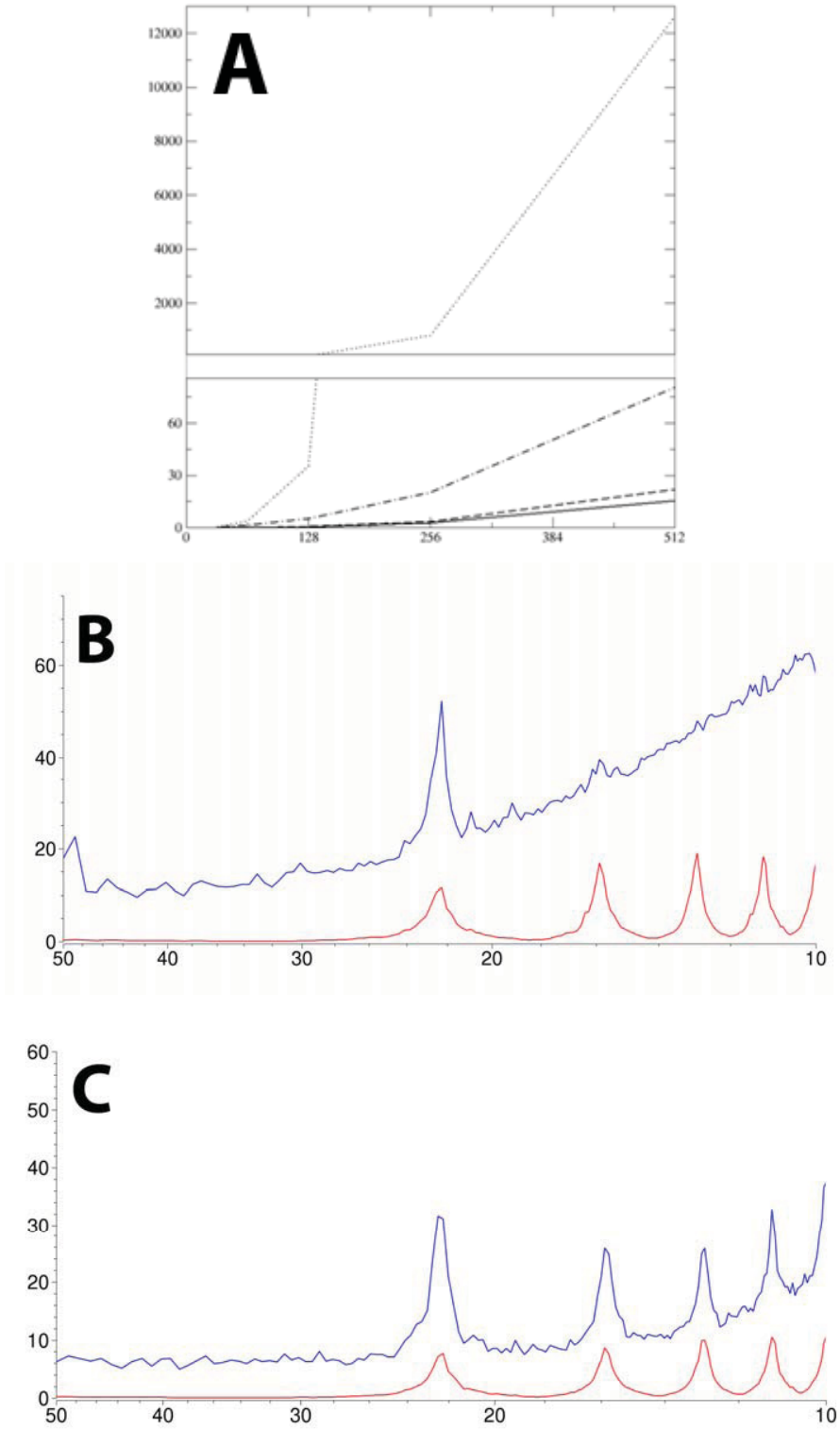
Each of the three algorithms available to apply the TCIF model represents a different equilibrium between speed and accuracy of the simulation. The Explicit Fourier summation algorithm allows a mathematically precise application of the model. However, the time needed to apply it scales as N^4 in the case of square images, where N is the length of the image edge. This makes its use in the size range of experimental images infeasible, as can be seen in Figure 6A. The Tilt Axis Rotation algorithm is the fastest and will also give a mathematically precise solution if the tilt axis is parallel to the y axis (Philippsen, 2007B), but will need a rotation of the image in all other cases. The interpolation operation needed to perform the rotation degrades the quality of the data, resulting in a remarkable loss of accuracy (See Figure 6B and C). The Shannon Interpolation algorithm is not as fast as the Tilt Axis Rotation, but the accuracy loss is not as severe as in the previous case. Availability of time and computational power and accuracy requirements are criteria that determine which algorithm to use to apply the TCIF model. It must be pointed out that the speed and accuracy data shown in Figure 6 for the Tilt Axis Rotation algorithm include the back-rotation that brings back the image to its original orientation.

FIGURE 7A: Comparison of the speed performance of the TCIF algorithms. The X axis represents the edge length of the square image to which the TCIF model is applied. The Y axis shows the time required to complete the operation (in seconds). The scale of the Y axis in the top part of the diagram has been compressed for ease of graphical representation.

FIGURE 7B and 7C: Comparison of the accuracy performance of the TCIF algorithms. Figure 5B and 5C show the comparison between Fourier Transforms of images generated using the Explicit Fourier summation algorithm and images created using the Tilt Rotation algorithm (in blue) and the Shannon Interpolation algorithm (in red). In both images the x axis shows the resolution. The y axis in Figure 5B shows the difference in amplitude (as a percentage of the amplitude computed by the Explicit Fourier algorithm). The y axis in Figure 5C shows the phase difference in degrees taking the phase computed by the Explicit Fourier algorithm as reference.

The Explicit Fourier summation simulation algorithm is mathematically exact and is used as a standard to evaluate the accuracy of the other algorithms. However, the way it scales with image size makes it unusable except for realistic-sized images. The other two algorithms represent different levels of trade off between speed and accuracy. The time and accuracy evaluations of the Tilt Axis Rotation algorithm include back-rotation, which is needed for comparison with the other algorithms, but usually not for day-to-day image-processing.

FIGURE 7


A
12000
10000
8000
6000
4000
2000
60
30
0
0
128
256
384
512
B
60
40
20
0
50
40
30
20
10
C
60
50
40
30
20
10
0
50
40
30
20
10

Chapter 2

Annex - Simulation and Correction of Tilted Weak-phase Samples

1 Introduction

This chapter will contain additional analysis and discussion on the material that was introduced in Chapter 1. That section of the manuscript was formatted for publication, and strict space limitations needed to be satisfied. Some interesting observations and some detailed descriptions of advanced topics where left out. They appear in this chapter. The first section of the chapter deals with film simulation protocols. Although the approach that was finally implemented in Chapter 1 is both efficient and realistic, a more detailed and physically realistic model was previously considered and discarded. The abandoned model is presented in the following paragraphs. Some feature and limitations of the protocol used in Chapter 1 are also discussed. The second section describes an attempt at the determination of the TCIF model parameters starting from a collected image. The third presents instead an attempt to invert the effect of the TCIF using a forward-fitting approach, very different from the inversion stratefy published in Philippsen et al. (2006) . Finally, a peculiar problem that emerged during the implementation of the TCIF model is discussed.

Throughout the discussion, IPLT algorithms that were developed to perform computational tasks are presented. When this happens, the text has the following format:

Algorithm Name (path) Algorithm description

The ”path” information is used to locate the source code of the algorithm in the IPLT main repository

2 Film Simulation

Because of its sample-dependent nature and contrary to the classical CTF, the TCIF cannot be expressed as a convolution in real space, and as a consequence, it is not possible to express it in Fourier space as the multiplication of the input function with a separate standalone function. This makes it impossible to draw plots which describe the modulation of the input signal in dependence of the frequency, as it is often done for the CTF. In order to have a graphical representation of its effects, the TCIF must be applied to an input (sample) function of some kind. A simulation of empty carbon film was chosen as generic input sample function in several experiments described in this manuscript. In addition to being sufficiently featureless to represent no specific sample in simulations, it is also almost always present as background in images of real-life biological samples. In order to create synthetic images of carbon film, it become imperative to define and implement an efficient simulation protocol. A physically realistic model was briefly considered, before implementation problems led it to be discarded in favor of a less complex but equally effective approach. While the simpler protocol is described in the Chapter 1 of this work, under the Film Simulation portion of the Methods and Implementation section, the physically realistic model and the reasons that led to its discarding are described and commented here.

Physically Realistic Model (Abandoned)

A physically realistic model that can can be used to generate simulated images of an arrangement of osmium atoms on a carbon film has been made available

in literature (Engel et al., 1974). The author considered a certain number of carbon and osmium atoms occupying a defined extent of space and the derived the equations describing their interaction with the incoming electrons. In order to restrict the protocol to the generation of a simple carbon film image, it was decided to follow the same approach, but consider only the carbon atoms. The implementation of this method never went beyond the initial stages, as the computation of one of the physical parameters required by the model, the complex electron scattering factor, turned out to be too onerous a task.

Model Description

For the sake of simplicity, a 1-dimensional version of the model is described. The principles presented here can however be easily extended to 2 dimensions.

To derive the model, the length of the carbon film is subdivided in cells of size:

$$\Delta x = \sqrt{\frac{C}{t\rho_c L_0}} \tag{2.1}$$

where C is the atomic weight of carbon, t is the film thickness, ρ_c is the specific weight of carbon and L_0 is Avogadro's number. Each cell is assumed to be occupied by a single carbon atom located at a (random) position δx_l with respect to the center of the cell x_l. With this arrangement, the (complex) amplitude transmittance $g(x)$ in real space is:

$$g(x) = u(x) + iFT^{-1}\left[F_c\left(\frac{\xi}{\lambda f}\right)\sum_{l=0}^{n-1} e^{\left(-2\pi i\xi\frac{x_l+\delta x_l}{\lambda f}\right)}\right] \tag{2.2}$$

where ξ is the coordinate in Fourier space, f is the focal length of the lens, $x_l = l\Delta x$ is the coordinate of the l^{th}elementary cell, $u(x_l)$ is an unscattered wave, $F_c$is the complex scattering amplitude of carbon and $FT^{-1}$is the inverse Fourier

transform. While most of the physical entities that appear in the expression are trivial to compute, the determination of the value of the complex scattering amplitude (or factor) presented some formidable challenges.

The Complex Electron Scattering Factor

When interacting with a solid, high energy electrons can be scattered elastically (without energy loss) or inelastically (with energy loss) (Cowley, 1992). While the optical effects of elastic scattering can easily be described using a scattering factor with a real value, the introduction of a complex factor is required in order to describe the effects arising from inelastic interactions.

Several physical phenomena where energy is exchanged are grouped under the generic label of inelastic scattering. A detailed discussion of their nature and of their contribution to the value complex scattering factor is beyond the scope of this manuscript (See for example Peng (1999)).

Precomputed and tabulated values for the complex electron scattering factors of most chemical elements are available in literature. In the paper which described the film simulation protocol, Haase (1970) is cited as the source. However, the cited paper restricts his analysis to low acceleration voltages (40 to 120 kV), far below the ones used in structural biology nowadays (200-300kV). Other available sources, such as Schafer et al. (1971) , share the same problem, dealing mostly with low energy electrons. Because of the large gap between the energy levels described in the papers and the ones required for modern biological applications, extrapolation from the published data was considered too risky. Sofware has been made available (Peng et al., 1996) to compute complex scattering factors for specific elements at an acceleration voltage of choice. However, the Debye-Waller factor, which describes attenuation of the scattering caused by thermal motion, must be provided. The value of the Debye-Waller factor depends on the

physical state (crystalline, amorphous) of the element, and quantum mechanics calculations are needed to compute it under the conditions required by the simulation algorithm. This task was considered too onerous for the scope of the project, and the detailed film simulation protocol was abandoned.

Boothroyd's protocol (Boothroyd, 2000) was in the end chosen to generate the simulated images presented in Chapter 1. The method avoids a detailed description of the physical phenomena involved in the interation of the carbon film with the electron beam, and instead opts for a empirical approach to image generation. The images that it can generate are however adequate and realistic-lloking.

Boothroyd's Film Simulation protocol

The description of Boothroyd's protocol was already given in the Methods and Implementation section in Chapter 1 of this work. It is not repeated here. The following paragraphs are dedicated to some in-depth analysis and discussion that did not fit in Chapter 1.

Shortcomings of Electon Microscopy Image Simulation Techniques.

Several computer simulation protocols have succeeded in generating images that qualitatively match high-resolution electron microscopy experimental images. A qualitative simulation has been regarded as successful when the computed image reproduced the general "look and feel" of a real micrograph. In recent years, however, the goal in the development of simulation protocols has shifted to the achievement of a quantitative match, where the intensity, the contrast and the pattern of a simulated image can be compared to the ones of an experimental micrograph. In these aspect, all currently available simulation methods fall short,

and Boothroyd's is no exception, as the author himself discusses in Boothroyd (1998) .

In particular, contrast seems to be the hardest parameter to reproduce faithfully. The average difference in contrast between a simulated and an experimental image has been estimated to be a factor of 3. This has been called the Stobbs factor (Hÿtch and Stobbs, 1994). The causes of the Stobbs factor are not precisely known although several sources have been suggested: inelastically scattered electrons (Herring, 2006), surface contamination (Boothroyd and Yeadon, 2003) and thermal diffuse scattering (Howie, 2004). Since the work described in this manuscript has no need for a full quantitative simulation of the carbon film, this problem was ignored.

It should be noted that extremely recent results still not published in a peer-reviewed journal but presented at an international meeting suggested an alternative and very different origin for the Stobbs factor: not unaccounted scattering phenomena but more simply the Modulation Transfer Function (MTF) of the recording device being ignored by the simulation protocols (Thust, 2008).

Analytical Form of the Boothroyd Film Simulation Protocol.

The Boothroyd simulation protocol lends itself gracefully to be expressed in an analytical form. As explained in Chapter 1 of this manuscript, the TCIF model is strongly sample-dependent. Its effect cannot be expressed as a simple multiplication of the Fourier transform of the input data with a standalone function (as happens with the classical CTF). Because of this, any analytical treatment of the model cannot be performed on a standalone function. A mathematical expression that incorporates the input data must be used as a basis. In the work described in this manuscript, the TCIF model is almost always applied to a film simulation. The availability of an analytical expression for the film simulation

allows then the full simulation (input data + TCIF model) to be expressed in a mathematical form.

In the TCIF model equation (Eq. 1 in Part 1 of this work), the input function appears in its Fourier transformed form. Because of this, in the following discussion an analytical expression for the film simulation in Fourier space is derived. The following treatment deals with a 1-dimensional film simulation. However, it can easily be extended to a 2-dimensional case. In its essence, a film simulation consists of a Poisson noise-filled image, weighted in Fourier space by the scattering factor of carbon. The characteristics of the Poisson distribution are determined by the average density of the carbon film area that falls within a pixel in the image. This can be translated into an analytical expression in a straightforward way. A discrete noise-filled real space image can be represented as a sum of Gaussians, each centered on the coordinates of the center of a pixel:

$$\phi = \sum_{n=1}^{N} \left(\frac{A_n}{2\pi\sigma} \right) e^{\frac{-(x-\mu_n)^2}{2\sigma^2}} \tag{2.3}$$

ϕ is the real space image, N is the number of pixels in the image, μ_n is the coordinate of the center of the n^{th} pixel, σ is chosen in such a way that the confidence limit of 3σ corresponds to the half the width of an image pixel, and A_n is the amplitude at the center of the Gaussian, which is a number randomly drawn from a Poisson distribution determined according to Boothroyd's protocol.

Due to the linearity property of the Fourier transform, the Fourier transform of a sum of functions is equivalent to the sum of the Fourier transforms of all functions in the sum. Since the Fourier transform of Gaussian function is still a Gaussian function, the Fourier transform of ϕ at a specific frequency ξ in Fourier space can be expressed in the following way:

$$\Phi(\xi) = \sum_{n=1}^{N} \left(\frac{A_n}{2\pi\sqrt{\sigma}} \right) e^{(i\xi\mu_n - \frac{\xi^2\sigma}{2})} \tag{2.4}$$

The only element missing to complete the analytical description of Boothroyd's protocol is the weighting according to the scattering factor of carbon:

$$\Phi(p) = F(\xi) \sum_{n=1}^{N} \left(\frac{A_n}{2\pi\sqrt{\sigma}} \right) e^{(i\xi\mu_n - \frac{\xi^2\sigma}{2})} \tag{2.5}$$

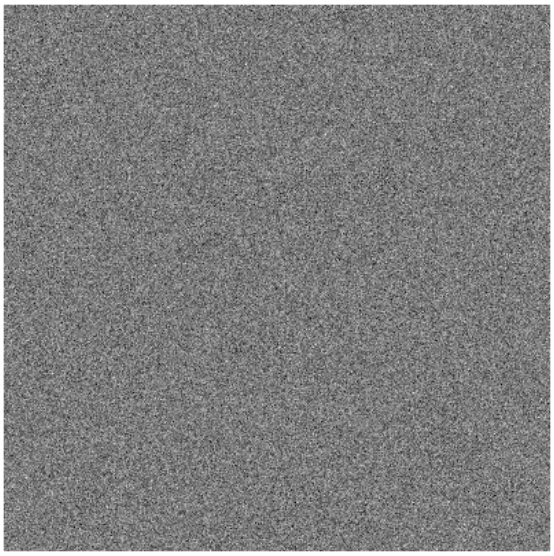
where $F(\xi)$ is the real electron scattering factor for carbon at frequency ξ , which can be easily recovered from the International tables for crystallography (Hahn, 2002).

This analytical expression is a full description (in Fourier space) of the Boothroyd film simulation protocol and can be plugged directly into the TCIF model described in Chapter 1.

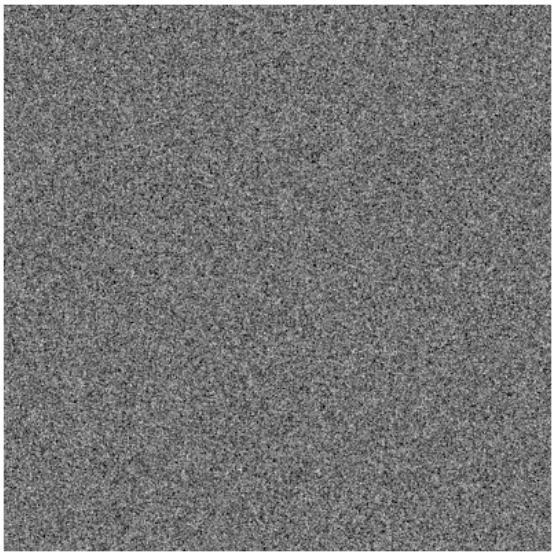
3 Fitting of the TCIF parameters

In order to correct and compensate for the effects of the TCIF on experimental electron microscopy data, it is necessary to know with sufficient precision the imaging condition under which the micrographs were collected. These conditions are described by the values of the parameters of the TCIF model, and their knowledge is necessary to perform reliably any type of correction.

Some of this parameters can be deducted directly from the instrument's setup and characteristics, since they are not strongly dependent on the quality of the instrument's alignment (for example wavelength of the incoming electrons or chromatic and spherical aberration constants). For others, however, like defocus distance or degree of astigmatism, a direct readout is often unreliable. The micrograph and its Fourier transform must then be analyzed computationally and the parameters must be deducted directly from the experimental data. When dealing with a classical CTF model on untilted images, this approach is common practice. Most image processing software packages designed for electron



Image



Plot of analytical function

Figure 2.1 Two real-space simulated film images generated using different approaches. The image on the left has been created using the image-based Film Simulation algorithm as implemented in IPLT. The image on the right has been obtained using the analytical representation of the film simulation described in the text. An empty Fourier transform of an image of the same size as the one on the left has been filled using the values of the analytical expression in correspondence with the center of each pixel. The image has then been back transformed to obtain the real-space version shown in the figure. Both simulations have a size of 400x400 pixels, a pixel width of 2 Angstroms, and the thickness of the simulated film is 100 Angstroms.

microscopy offer some kind of graphical interface that allows the user to fit some parameters to the power spectra of images in in an interactive fashion (For example IPLT (Philippsen et al., 2007), EMAN (Tang et al., 2007), SPIDER (Shaikh et al., 2008), Bsoft (Heymann and Belnap, 2007)). Other software packages, like CTFFIND and CTFTILT by N. Grigorieff (Mindell and Grigorieff, 2003)or Xmipp (Sorzano et al., 2007) offer their users the opportunity to automatize the process completely, requiring merely a rough starting estimation of the parameter's values. Not only this latter approach is more comfortable to the user, but it also helps removing human bias from the process, giving more reproducible results.

In order to establish a successful image processing protocol for tilted images, similar tools based on the TCIF model must be made available. As a first approach, work was started on a technique aimed at fitting all the parameters of the TCIF model to the experimental data, including the ones that can be reliably read out from the instrument. The procedure was however designed to allow the exclusion of some of the parameters from the fit, in case trusted estimations were available for them. The overabundance of data, compared to the low numbers of parameter being fitted, suggested a reduction of the dataset size before the start of the procedure. To perform the fitting itself, the Levenberg-Marquardt algorithm was chosen. This section of this manuscript describes the work that has been done, describing in detail the data reduction step, and the mathematical background of the fitting algorithm. Additionally, the results of the first preliminary tests are reported and discussed.

Reduction of the Data Set

The first issue that had be faced in in order to develop an efficient fitting algorithm for the TCIF model was the overabundance of data. A typical micrograph

is digitalized on a grid spanning from least 2000 x 2000 pixels (for electron tomography) up to 8000x8000 and more (electron crystallography). The Fourier transform of the micrograph image, which appears in the TCIF model, contains then from 4000000 to 64000000 data items. Performing any kind of fitting procedure on a dataset of such huge dimensions would be extremely time consuming. One must remember, however, that even assuming that all the parameters in the TCIF model take part in the fitting, only 7 values must be determined. With such a high data elements to number of unknowns ratio, the system is strongly overdetermined. The dataset can be then conveniently reduced to increase the computational speed, without worrying too much about the impact on the stability of the procedure.

One way to decrease the size of the dataset in a useful way would be to average redundant data in order to increase the signal to noise ratio. Averaging is often performed on the data when dealing with a classical CTF model. For example, real space images are often disassembled in tiles, and tiles for which the values of the parameters are predicted to be similar are averaged (Frank, 1996).

This approach relies on the fact that the CTF model sees the image and its Fourier transform as the convolution or the multiplication of the sample function with a separate stand-alone function. The standalone function depends on the parameters than need to be fit and their values can be deducted from its shape. When the tiles are averaged, the local sample function, which is different for each tile, gets averaged out, while the standalone function gets reinforced. The values of the parameters can then be fitted to it. However, this concept is inapplicable to the TCIF model. In the TCIF model there is no such thing as a separated standalone function, and the relationship between the image and the sample function cannot be described by a simple mathematical operation. The sample function information needs to be preserved, and the kind of averaging that has just been described destroys it.

Another technique often used when dealing with the CTF model is the radial averaging of power spectra of astigmatism free-images. Experimental astigmatism-free images are rare, but astigmatism can be computationally removed during the image processing, assuming that parameters describing it are available. Unfortunately, the procedure being developed aims at being able to fit all the parameters, including the ones describing the astigmatism, which are then not available. It should be however pointed out that the property of rotational symmetry of the power spectrum is also valid for astigmatism-free tilted images.

Since averaging is not possible, a reduction of the data can only be performed by selecting a subset of the data and discarding the rest. The minimum size of the dataset that still allows convergence should be rigorously determined. This step was however left for a later time During the early development of the fitting procedure, the size of the reduced dataset was determined empirically: only information coming from the two main axes and the two diagonals of the power spectrum were kept. This was intuitively considered enough to adequately sample the power spectrum of the titled image in its radial and angular extent. However, if the data reduction strategy reveals itself to be inadequate, it can always be changed and refined in the future.

Extract1DData (valerio/tcif/lib/extract_1d_data.hh) This is a helper algorithm that extracts data from specific parts of the Fourier Transform of an image and makes them available to the user in the form of vectors of complex numbers. Once the algorithm is applied to a half-frequency image, it is possible to retrieve data from the main axes and the two diagonals, using themethods that the algorithm provides

Parameter Fitting with the Levenberg-Marquardt algorithm

The fitting of the parameter to the reduced dataset is performed using a procedure

based on the Levenberg-Marquardt algorithm (Marquardt, 1963). As several other fitting algorithms, this technique minimizes a least-square merit function that describes how well a mathematical model correlates with the data. It is designed to combine the best advantages of two popular algorithms used in non-linear function minimization, the steep descent method and the inverse-Hessian method (Polak, 1971), by varying smoothly between them. In particular, the latter is used far from the minimum of the merit function, and a continuous switch to the former happens as the minimum is approached. Being efficient and easy to implement, this method has become prominent in non-linear least-square modeling and appeared to be the natural choice as the base of the TCIF parameter fitting algorithm. The free and open source Gnu Scientific Library (2009) offers an implementation of the Levenberg-Marquardt algorithm, which was used during the implementation of the fitting procedure

In order to use the Levenberg-Marquardt algorithm, an analytical model of the data must be built. Derivatives of the model with respect to the parameters that need to be determined must also be made available. Since the parameters of the TCIF model were fitted to the power spectrum of a tilted image, an analytical model of the spectrum itself was derived. The main difficulty came from the fact that the TCIF model needs to be applied to a sample function, and the sample function for an experimental image is obviously unknown. Actually, the final goal of the whole image processing effort is to retrieve it. Images of biological samples in electron microscopy are however often collected after the sample is deposited on a carbon film. Because of this the film appear as background in experimental images, and the sample function generated by the biological sample is always overlaid on the one generated by the carbon film. It was then decided to use the latter as a "generic" sample function. A mathematical expression for the carbon film simulation has been derived in subsubsection 2.2.2.2 and can be plugged into the TCIF model directly.

Given the TCIF model:

$$Q(\boldsymbol{p}) = i\left(\Phi(\boldsymbol{p} - \frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha)e^{-iW_0(\boldsymbol{p})} - \Phi(\boldsymbol{p} + \frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha)e^{iW_0(\boldsymbol{p})}\right) \tag{2.6}$$

The Fourier transform of the sample function at the frequency $\boldsymbol{p} - \frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha$ can be written as:

$$\Phi(\boldsymbol{p} - \frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha) = F(\boldsymbol{p} - \frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha)\sum_{n=1}^{N}\left(\frac{A_n}{2\pi\sqrt{\sigma}}\right)e^{(i\left(\boldsymbol{p}-\frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha\right)\boldsymbol{\mu_n}-\frac{\xi^2\sigma}{2})} \tag{2.7}$$

a similar expression can be written for the Fourier transform at the frequency $\boldsymbol{p} + \frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha$:

$$\Phi(\boldsymbol{p} + \frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha) = F(\boldsymbol{p} + \frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha)\sum_{n=1}^{N}\left(\frac{A_n}{2\pi\sqrt{\sigma}}\right)e^{(i\left(\boldsymbol{p}+\frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha\right)\boldsymbol{\mu_n}-\frac{\xi^2\sigma}{2})} \tag{2.8}$$

Both values are complex numbers. If we write:

$$\Phi(\boldsymbol{p} - \frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha) = a + ib \tag{2.9}$$

$$\Phi(\boldsymbol{p} + \frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha) = c + id \tag{2.10}$$

The TCIF model becomes:

$$Q(\boldsymbol{p}) = i(a + ib)e^{-iW_0(\boldsymbol{p})} - i(c + id)e^{iW_0(\boldsymbol{p})} \tag{2.11}$$

The model must be fitted to the power spectrum of the experimental image. The squared modulus of $Q(\boldsymbol{p})$, is equal to:

$$|Q(\boldsymbol{p})|^2 = a^2 + b^2 + c^2 + d^2 - 2(ac + bd)\cos(2W_0(\boldsymbol{p})) + (-2bc + 2ad)\sin(2W_0(\boldsymbol{p})) \tag{2.12}$$

a,b,c and d can be written as:

$$a = \xi_{minus} R_{cm} \tag{2.13}$$

$$b = -\xi_{minus} R_{sm} \tag{2.14}$$

$$c = \xi_{plus} R_{cp} \tag{2.15}$$

$$d = -\xi_{plus} R_{sp} \tag{2.16}$$

$$\xi_{minus} = e^{-\pi^2 \sigma^2 p_{minus}^2} \tag{2.17}$$

$$\xi_{plus} = e^{-\pi^2 \sigma^2 p_{plus}^2} \tag{2.18}$$

$$R_{cm} = \sum_{n=1}^{M} \cos\left(2\pi \boldsymbol{\mu_n} \cdot \boldsymbol{p_{minus}}\right) A_n \tag{2.19}$$

$$R_{sm} = \sum_{n=1}^{M} \sin\left(2\pi \boldsymbol{\mu_n} \cdot \boldsymbol{p_{minus}}\right) A_n \tag{2.20}$$

$$R_{cp} = \sum_{n=1}^{M} \cos\left(2\pi \boldsymbol{\mu_n} \cdot \boldsymbol{p_{plus}}\right) A_n \tag{2.21}$$

$$R_{sp} = \sum_{n=1}^{M} \sin\left(2\pi \boldsymbol{\mu_n} \cdot \boldsymbol{p_{plus}}\right) A_n \tag{2.22}$$

$$\boldsymbol{p_{minus}} = \boldsymbol{p} - \frac{1}{2} p^2 \lambda \boldsymbol{d} \tan\alpha \tag{2.23}$$

$$\boldsymbol{p_{plus}} = \boldsymbol{p} + \frac{1}{2} p^2 \lambda \boldsymbol{d} \tan\alpha \tag{2.24}$$

It is possible at this point to write a full analytical expression for the power spectrum of a simulated tilted image of a carbon film:

$$\begin{aligned} |Q(\boldsymbol{p})|^2 &= E_a R_{cp}^2 + E_a E_b^2 \left(R_{cm}^2 + R_{sm}^2\right) + E_a R_{sp}^2 - \\ &\quad 2E_a E_b \left(R_{sm}\left(W_{sin} R_{cp} + W_{cos} R_{sp}\right) + R_{cm}\left(W_{cos} R_{cp} - W_{sin} R_{sp}\right)\right) \end{aligned} \tag{2.25}$$

Where:

$$E_a = e^{-\frac{1}{2}\pi^2 p^2 \sigma^2 \left(4 - p^2\lambda^2 + p^2\lambda^2 \sec\alpha^2 + 4\lambda \boldsymbol{p}\cdot\boldsymbol{d}\tan\alpha\right)} \tag{2.26}$$

$$E_b = e^{2\pi^2 p^2 \lambda \sigma^2 \boldsymbol{p}\cdot\boldsymbol{d}\tan\alpha} \tag{2.27}$$

$$W_{sin} = \sin\left(2W_0(\boldsymbol{p})\right) \tag{2.28}$$

$$W_{cos} = \cos\left(2W_0(\boldsymbol{p})\right) \tag{2.29}$$

Its derivatives with respect to all the parameters can be computed easily. The complexity of the model, however, causes the expressions to be long and intricate. To avoid breaking the flow of the discussion, the derivatives are not fully shown in this chapter. Although some of their features are introduced and discussed in the following paragraphs, their extended expressions can be found in Appendix 1.

The Levenberg-Marquardt algorithm requires an initial estimation of the parameter values. During the first tests, synthetic data, for which the precise values of the parameters are known, were used. Since the focus of the analysis was on the stability and the convergence of the algorithm, the precise parameter values were used as starting conditions, introducing a perturbation of random magnitude and sign to simulate uncertainty in their estimation. Modulation of the magnitude of the perturbation was later used to explore the solution space and determine the convergence range of the algorithm.

AnalyticalAbsQ (valerio/tcif/lib/analytical_abs_Q.hh) This algorithm is used to compute the value of the power spectrum of a simulated carbon film tilted image at a specific spatial frequency $\boldsymbol{p}$, using the analytical model described in the previous paragraph. It can also be used to compute the values of the derivatives with respect to all the parameters at the same spatial frequency. The algorithm must be initialized with the size of the image being simulated and its pixel width, together with the values of the model parameters and the thickness of the simulated carbon film. The algorithm

can be then called while specifying a particular spatial frequency. A structure containing the values of the power spectrum and of all the derivatives is returned (AbsQStruct). Internally the algorithm is optimized for speed. Short sequences of operations that repeatedly appear in the mathematical expressions are precalculated. This approach avoids multiple computations of the same terms, increasing the operational speed of the algorithm. Detailed analytical forms of the precalculated terms are available in Appendix 2.

First Tests: Comments and Insights

A complete test of the parameter fitting algorithm could not be completed. Several issues emerged during the preparation stage. Even though the algorithm that computes the analytical expressions has been carefully optimized, its speed is still not high enough to allow its application to realistic sized images. In preliminary tests, when using a simulated image of 1000 by 1000 pixels, much smaller that the typical size of scanned micrographs, the computation of the power spectrum and all derivatives for a single spatial frequency took several minutes. In order to be usable for realistic experimental data, the algorithm must not only be able to work with images that can be up to 64 times bigger, but also to compute the required values for million of different spatial frequencies in each single fitting step. This level of scalability is currently completely out of reach.

Furthermore, the behavior of the parameter derivatives does not lend itself very well to the use of a fitting procedure. Figure 2.3 shows plots of the derivatives with respect to the parameters over a wide range of parameter values. The plots were created by assuming a simulated image size of 1000 by 1000 pixels with a pixel width of 2 Angstrom. Typical imaging parameters for a 60 degrees tilted image were used. The parameters were then changed one a a time. For each parameter the vicinity of the starting value in parameter space was explored. The figure reports the behavior of all the derivatives for a point at 80% of the Nyquist frequency, as an example. It can be seen that several of the derivatives (in particular the ones related to the Defocus parameter, to the Amplitude Contrast magnitude, to the Chromatic Aberration constant and to the direction of the tilt angle) show oscillating behavior with several local minima. These minima can create problems to any fitting algorithm that relies on the minimization of a score function, like the Levenberg-Marquardt. To avoid the risk of getting stuck into them, the starting estimations of the parameter values would have to be so close to the correct final ones that the algorithm would only be useful for the

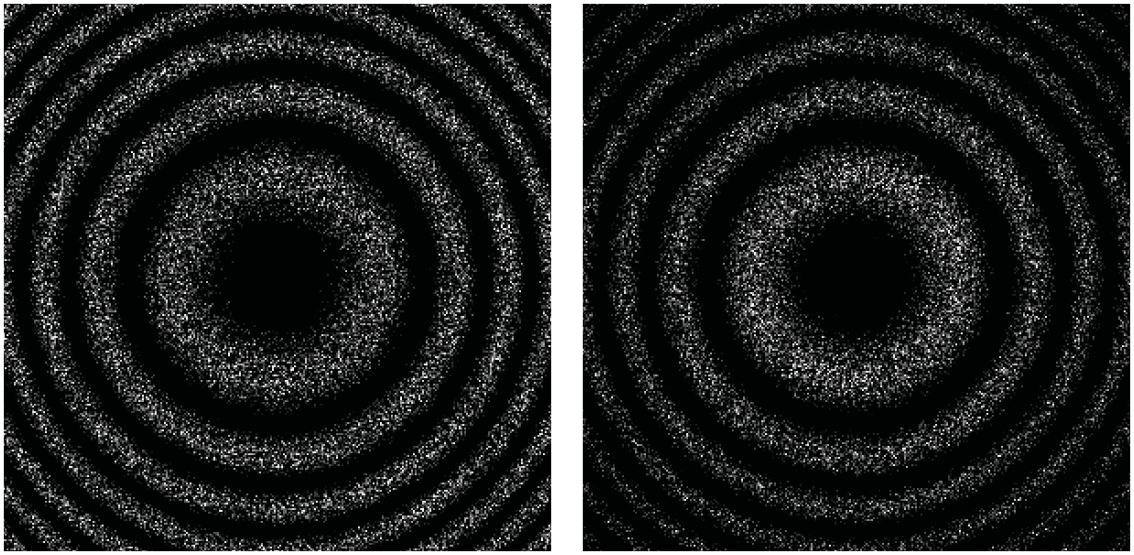


Image Plot of analytical function

Figure 2.2 Power spectra of simulated untilted images of a carbon film. The image-based IPLT algoritm was used to generate the image on the left. The image on the right was generated as a plot of the analytical expression described in the text. Imaging conditions: image size 1000x1000 pixels, pixel width: 2 Angstroms, defocus value: 2 micrometers.

refinement, and not for the initial determination of the experimental imaging conditions.

Outlook and Future Development

In order to make the Levenberg-Marquardt-based fitting algorithm viable for use with realistic-sized images, its computational speed must be increased strongly. Several approaches are possible. Since the calculations for each spatial frequency is independent from the others, and all rely on a simple common set of parameters, parallelization is surely a strategy that could be followed.

Unfortunately, the software framework in which the method has been currently implemented, IPLT, does not support parallelized computing. A complete reimplementation of the algorithm would be necessary, and this makes the search for an alternative strategy preferable. The analysis of figure 2.3, can give useful hints. It is remarkable that, although the analytical expressions for the derivatives are very complex, their behavior appear to follow simple well-known schemes. For example, the derivatives with respect to the Chromatic Aberration constant, to the Amplitude Contrast magnitude and to the Tilt Angle show the behavior of modulated oscillating functions. The derivative with respect to the defocus appears to follow a simple oscillation, while the derivative with respect to the Wavelength features an exponential-like rise. Finally, the derivative according to the Astigmatism angle appears to behave linearly. All this suggests that although the derivative expressions are very complex, some simple terms are probably dominant over the rest and dictate the large scale behavior of the function. This situation could be exploited by identifying the dominant terms and simplifying the derivative expressions, neglecting the less important terms. This would speed up computational time enormously while retaining a sufficient accuracy.

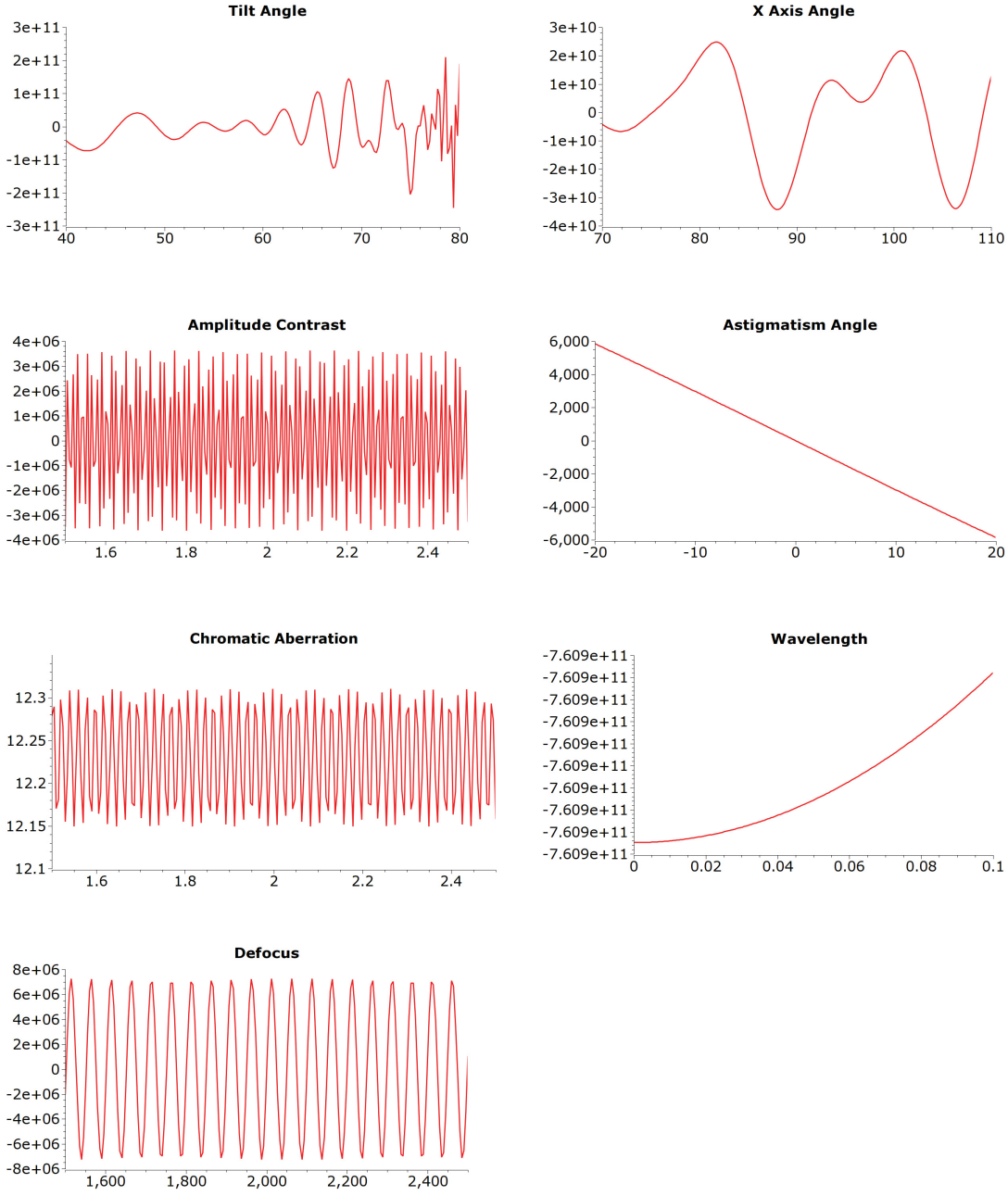


Figure 2.3 Examples of derivatives of the Film Simulation + TCIF model with respect to all the parameters. The values of the derivatives were computed starting from a set of imaging conditions, varying the conditions one at a time, exploring the parameter space in the surrounding of the starting value, and computing for each case the value of the derivative. Starting conditions: tilt angel 60 degrees, tilt axis angle with respect to the x axis 90 degrees, amplitude contrast 0.07, astigmatism angle 0 degrees, spherical aberration 2 mm, wavelength 0.25 picometers, defocus value 2000 nm. Image size: 1000x1000. Pixel width: 2 angstroms. Spatial frequency for which the derivatives were computed: 80% of the Nyquist frequency.

Both these methods do not solve the local minima problem. It is in the end possible that a full-scale parameter optimization approach is not worth the computational effort, and that the values of the imaging parameters are better determined one at a time using established heuristic methods that not only have proved themselves to be sufficiently accurate for basic electron crystallography image processing, but are also under constant development.

4 Inversion of the TCIF

It has been shown that the TCIF model allows the recovery of the original sample function from an experimental collected image, provided that the the imaging conditions are known. In particular, it can be shown that when the direction of the tilt axis is parallel to the y axis of the reference system, the problem reduces to a system of linear equations of the form $Ax = B$, where the known matrix B represents the Fourier Transform of the experimental image, the unknown matrix x represents the Fourier Transform of the sample function, and the coefficients of the matrix A are determined by the parameters of the TCIF model. Results have been published where the system has been solved for small test images using Single Value Decomposition, with a full recovery of the original sample function (Philippsen et al., 2006). However, this approach does not scale well: the size of the linear system increases quickly with the size of the observed image , and solving it for realistic images can take days on a common desktop computer. This strong limitation prompted the search for an alternative strategy and led to the attempt of solving the problem using a forward-fitting technique. An initial estimation of the sample function was used as a starting point. The TCIF mathematical model was applied to it and the result was compared with the final observed image. The difference between the two was quantified, and over several iterations, the starting sample function was altered in a way that minimized the difference. The three requirements for an approach of this kind are: an analytical expression for the TCIF model, which is available, an algorithm that can minimize a scoring function, and a rough estimation of the parameter values to initiate the minimization.

Forward-fitting TCIF Inversion Algorithm

A natural choice for the minimization algorithm was the Levenberg-Marquardt technique described previously (Marquardt, 1963). The algorithm needs a model

of the the data, which is provided by the TCIF model. The unknown terms are the values pixels of the original sample function image. The derivatives of the model with respect to all their values must then be made available. In the following paragraphs the procedure followed to derive all the needed analytical expressions is described and commented.

The TCIF model has the following form:

$$Q(\boldsymbol{p}) = i\left(\Phi(\boldsymbol{p} - \frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha)e^{-iW_0(\boldsymbol{p})} - \Phi(\boldsymbol{p} + \frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha)e^{iW_0(\boldsymbol{p})}\right) \tag{2.30}$$

where Q is the Fourier Transform of the collected image and Φ is the Fourier Transform of the original sample function. A detailed description of all the other parameters can be found in Chapter 1 of this manuscript. When the model is applied to a discretized Fourier Transform, and $\boldsymbol{p}$ corresponds exactly to one of the sampled frequencies, $(\boldsymbol{p}-\frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha)$ and $(\boldsymbol{p}+\frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha)$ do not generally correspond to other sampled spatial frequencies. Thanks to the Shannon theorem (Henderson et al., 1949), however, whose conditions are satisfied, the value of Φ at a generic spatial frequency $\boldsymbol{p}$ can be written as a function of all the values in the discretized dataset:

$$\Phi(\boldsymbol{p}) = \sum_{n=0}^{N_n} \Phi(\boldsymbol{p_n})Sinc(\boldsymbol{p} - \boldsymbol{p_n}) \tag{2.31}$$

N_n represent the number of samples in the dataset, $\boldsymbol{p}_n$ is the spatial frequency of sample n, and $Sinc$ is the following 2-dimensional Sinc fuction:

$$Sinc(\boldsymbol{p}) = \frac{\sin(\frac{1}{w}\pi p_x)}{\frac{1}{w}\pi p_x}\frac{\sin(\frac{1}{w}\pi p_y)}{\frac{1}{w}\pi p_y} \tag{2.32}$$

(w is the reciprocal space pixel width).

Considering this result, and separating the complex values of $\Phi(\boldsymbol{p}_n)$ in their amplitude and phase parts (A_n and γ_n), the TCIF model can be rewritten as:

$$R_n = \sin\left(W_0(p_n) - \gamma_n\right) Sinc\left(|\boldsymbol{p_{nminus}} - \boldsymbol{p_n}|\right) + \sin\left(W_0 + \gamma_n\right) Sinc\left(\left|\boldsymbol{p_{nplus}} - \boldsymbol{p_n}\right|\right) \tag{2.33}$$

$$I_n = \cos\left(W_0(p_n) - \gamma_n\right) Sinc\left(|\boldsymbol{p_{nminus}} - \boldsymbol{p_n}|\right) + \cos\left(W_0 + \gamma_n\right) Sinc\left(\left|\boldsymbol{p_{nplus}} - \boldsymbol{p_n}\right|\right) \tag{2.34}$$

Where:

$$\boldsymbol{p_{nminus}} = \boldsymbol{p_n} - \frac{1}{2} p_n^2 \lambda \boldsymbol{d} \tan\alpha \tag{2.35}$$

$$\boldsymbol{p_{nplus}} = \boldsymbol{p_n} + \frac{1}{2} p_n^2 \lambda \boldsymbol{d} \tan\alpha \tag{2.36}$$

$$Q(\boldsymbol{p}) = \sum_{n=0}^{N_n} A_n R_n + \sum_{n=0}^{N_n} A_n I_n \tag{2.37}$$

In this form, the TCIF model was used in the Levenberg-Marquadrt algorithm. The derivatives with respect to all the pixels in the discrete original sample function (in their amplitude and phase form), can easily be derived. They are not reported in this chapter to avoid breaking the flow of the discussion, but their full analytical expressions can be found in Appendix 3.

During the first tests of the inversion algorithm, synthetic data were used. The precise sample function used to generate them was known. A perturbed version of the same function was used as starting estimation of the values of the unknown terms. The perturbation, introduced to simulate the uncertainty of realistic a starting condition, was of random magnitude and sign. The average magnitude of the perturbation was initially kept small, but was increased in subsequent tests. The goal was to analyze the convergence properties of the algorithm as the accuracy of the initial prediction deteriorated.

TCIFFit (valerio/tcif/lib/tcif_fit.hh) This algorithm is based on a 1-dimensional version of the model described in the previous paragraphs. The direction

of the tilt axis is assumed to be parallel to the y axis of the reference system, and the Fourier Transform of the observed image is processed one horizontal line at time. The complex values of the line are provided to the algorithm at initialization, together with the x coordinate of the first pixel of the line and with the values of the parameters describing the imaging conditions. The algorithm needs then a starting estimation of the discrete values of the sample function, which are provided in the form of another series of complex numbers. The minimization can then be started. When the processing is complete, the solution is returned once more in the form of a series of complex numbers. Internally, the algorithm performs the fit using the complex values of the sample function in the form ”amplitude + phase”, and the complex values of the final observed image in the form ”real part + imaginary part". The values of the interpolating Sinc functions are precomputed whenever possible, and all the derivatives are evaluated in a single step.

First Tests: Comments and Insights

The first tests of the inversion algorithm showed a very limited convergence range. When the sample function provided as initial estimation was exactly the same as the one used to generate the synthetic data, the algorithm correctly identified it as perfect solution. However, when the pixels in the initial estimation were subject to a perturbation of just 1% of their absolute values or phases, the algorithm converged to the wrong solution. Although a detailed analysis of the solution space has not been carried out, the source of problem is likely to be the presence of a number of local minima in the vicinity of the correct solution. The algorithm seems unable to differentiate them from the global minimum. It should also be noted that the tests were performed using exactly the same imaging parameters that were used to create the data. This condition is not realistic: although proce-

dures are available to determine the imaging parameters with sufficient accuracy, their estimation is likely to contain some error. Due to the very poor convergence of the algorithm even in the best conditions, the effect of this error has not been investigated.

Outlook and Future Development

The attempt to invert the effects of the TCIF using a Levenberg-Marquardt-based forward-fitting algorithm proved to be unsuccessful. Until some solution to the problem of the local minima is found, this technique appears not worth developing.

It should be pointed out, however, that in its original $Ax = B$ form, the inversion problem belongs to a family of cases that have been thoroughly investigated by mathematics in the past. Specifically, even in the smaller field of image processing, the challenge of solving massive systems of linear equations with thousands of parameters is hardly unknown. In medical tomography and even in biological imaging, approaches like the Algebraic Reconstruction Techniques (ART) (Natterer, 1986), routinely deal with similar systems. This techniques are well established and several optimized algorithms and implementations are available. Some of them, like the Simultaneous Algebraic Reconstruction Techniques (SART) (Kalarat et al., 2005) are designed for parallel computing. Although the current software environment, IPLT, does not support parallelization, the ability to solve the system using several computers and multiple CPUs could pave the way for an efficient inversion of the TCIF model with a complete recovery of the original sample function even for realistic-sized images.

Recently, a new trend emerged in the field of image processing for structural biology techniques, the use of GPUs, specialized processors present on commercial graphic cards available for any model of personal computer, to perform

calculations with a relevant increase in speed compared to the common CPUs (Castano-Diez et al., 2008). This is possible because GPUs are optimized to perform vector calulations, a requirement for the 3-dimensional graphic capabilities that the cards provide. The use of GPUs could provide a way to perform the complex calculations required by the inversion of the TCIF model on images of realistic size in an acceptable amount of time. However, graphic cards have a limited amount of memory, used to store just small graphic elements that must be applied to 3-dimensional images. This memory is usually too small to hold the big images that are used by the inversion algorithm. An approach that consists in the splitting of the image in small tiles which are processed independently is prevented by the nature of the TCIF model. The inversion is based on a system of the kind:

$$Ax = B \tag{2.38}$$

For an image of size n, each equation in this system has n unknowns. It is obvious that for the system to be determined, it must contain n equations. In other words, the inversion must be performed by considering the whole image at the same time, and a subdivision into tiles is not possible. This only means that a direct inversion of the TCIF model cannot be performed on the GPUs of the graphic card. Other approaches, however, like the SART technique described above, could reveal to be easier to implement.

5 The 0-line anomaly

Chapter 1 of this manuscript displays several simulations of power spectra of tilted images. These simulations were created by writing software implementations of the TCIF model and by applying them to synthetic sample functions. All these implementations presented in CHapter 1 share a peculiar characteristic. When the model is applied to a generic input image and the power spectrum of the resulting simulation is analyzed, the pixels of the line corresponding to spatial frequency 0 along the y axis appear to contain incorrect values. While the lines immediately above and below appear correct and blend smoothly with the rest of the simulation, the 0-frequency line represents an exception. This anomaly has been toroughtly investigated, and displays interesting features. The anomaly only appears when tilted images are simulated. When the tilt angle is set to 0, it cannot be detected. Its appearance is independent from the choice of TCIF model implementation. Even though the different implementations choose vastly different strategies to apply the model, the anomaly is always present. Curiously, it even appears when images are generated using the analytical expression described in chapter section 2.3.2. In this case, the anomaly shows itself when values corresponding to a y spatial frequency of exactly 0 is computed. If an amplitude plot of the 0-line is generated by applying the analytical expression to a discretized dataset, one can see that the values corresponding to sampled spatial frequencies are correct, but that values between them are subject to wild oscillations whose amplitude surpasses the average amplitude of the image. When a tilted image is simulated, the TCIF model uses these values, and this generates the anomaly. Since the anomaly appears even when using the analytical expression, its origin appears not to line in an implementation problem. The anomaly appears indeed to ba a feature of the TCIF model itself. Its source is not known. However, it should be noted that during the derivation of the tcif model, several constant terms are grouped together and simplified from the equations (Philippsen et al.,

2006). It is possible that one of these terms assumes a special value in correspondence with the 0 frequency line, and should not be simplified. An empirical solution to the problem was found. When the mean value of the input image in real space is 0, the anomaly disappears. While the reason for this is unknown and probably tied to the nature of the terms discussed above, this solution has been used to generate all the simulations in Chapter 1. The values of the pixels of the input images are always rescaled in such a way that the mean of their values is 0. The origin and the characteristic of the 0-line anomaly surely deserve some more investigation and analysis.

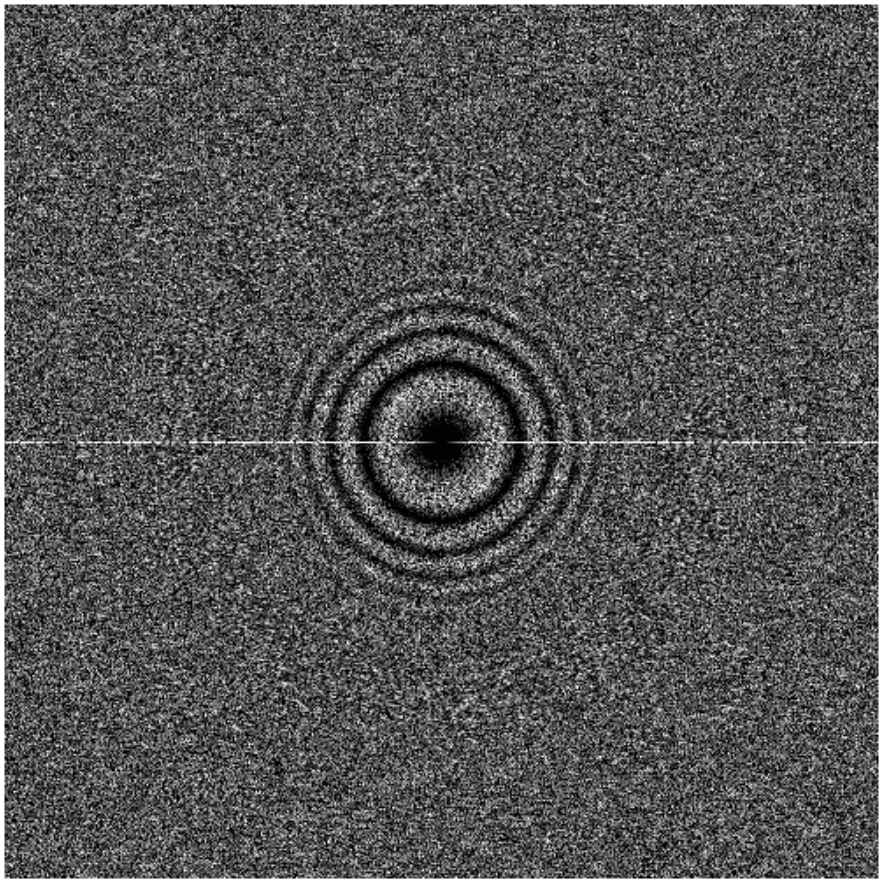


Figure 2.4 Power spectra of a simulated tilted image of a carbon film, showing the 0-frequency line anomaly

Chapter 3

On the correction of tilted 2D crystals

The next pages of this manuscript contain the draft of a journal article. The page, equation and figure numberings of the article have been preserved for an easier readibility. The article misses the introduction and it constantly refers to the TTF model and the TTF correction scheme. Their derivation and descriptions can be found in Henderson et al. (1986) and Henderson et al. (1990)

On the Correction of Tilted 2D Crystals

Valerio Mariani, Andreas D. Schenk, Andreas Engel and Ansgar Philippsen

Maurice E. Müller Institute of Structural Biology, Basel

Corresponding Author: **ansgar.philippsen@unibas.ch**

Abbreviations and Symbols

TCIF Tilted Contrast Imaging Function

CTF Contrast Transfer Function

FT Fourier transform

PSF Point Spread Function

α tilt angle of specimen out of plane

β orientation of tilt axis within image plane, β=0 means tilt axis is along y-axis

$\mathbf{C_s}$ Spherical aberration constant of instrument

Δ Sampling distance in digitized image

λ Electron wavelength, calculated from acceleration voltage

N Number of pixels in digitized image

$\mathbf{z_0}$ Base defocus (defocus at center of specimen)

χ_0 Reference angle for astigmatism

χ Astigmatism angle

$\mathbf{z_a}$ Defocus difference due to astigmatism

$\mathbf{W_0}$ Scherzer formula at the base defocus

Q Transformation describing tilted contrast imaging

Å 10^{-10}m

Introduction

In this publication, we focus on the differences between the previously described correction strategy, based on the TTF formalism, and our insight into the imaging process of 2D crystals under tilted conditions, derived from the TCIF.

Instead of recapitulating the TTF formalism at this point, we choose the alternative route in demonstrating how this formalism can be derived from approximations in the TCIF. We first do this in a graphical way, and then based on the previously published derivation.

We deliberately separate the image formation and the subsequent correction. This is necessary since the suggestion correction mechanism based on the TTF would be correct if the imaging process could be represented by the TTF to begin with, something that we wish to question in this paper. In other words, we first discuss the TTF as an approximation to the imaging process, and then discuss the correction scheme involving the TTF.

Background

In our first work on the tilted contrast imaging function (TCIF, Philippsen et al 2007B), we have derived a concise mathematical description of the imaging process for weak-phase samples under tilted conditions. We have shown that this imaging process cannot be described by a convolution, and hence removing the resulting optical artifacts from the images is more involving than for non-tilted images, where the contrast transfer function (CTF) is valid. In a proceeding publication (….), we have put this theoretical framework into practical use, first by demonstrating that we can predict the TCIF effect by matching experimental tilted carbon film images with equivalent simulations, and subsequently utilizing this predictive power to evaluate stripe-based correction schemes for electron tomography datasets.

The previously derived mathematical description of the TCIF is given by

$$Q(\mathbf{p}) = i\left\{e^{-iW_0(\mathbf{p})}\,\Phi\left(\mathbf{p} - \tfrac{1}{2}\mathbf{d}p^2\lambda\tan\alpha\right) - e^{iW_0(\mathbf{p})}\,\Phi\left(\mathbf{p} + \tfrac{1}{2}\mathbf{d}p^2\lambda\tan\alpha\right)\right\}. \qquad (1)$$

Here, Q is the resulting measurement in Fourier space, Φ is the Fourier representation of the sample, $\mathbf{p}$ is the 2D frequency vector parallel to the image plane, $\mathbf{d}$ is a unit vector perpendicular to the tilt axis, and α is the tilt angle. The term W_0 is based on the Scherzer formula (Scherzer, 1949), and is given by

$$W_0(\mathbf{p}) = \frac{\pi}{2} C_s \lambda^3 p^4 - \pi \lambda z_0 p^2 + \frac{\pi}{2} z_a \lambda p^2 \sin(2\chi - \chi_0), \tag{2}$$

where C_s is the spherical aberration of the microscope, λ is the electron wavelength at the microscope acceleration voltage, z_0 is the base defocus in the center of the sample, z_a is the defocus difference due to astigmatism, χ_0 is the reference angle used to describe the direction of the astigmatism, and $\chi = \tan^{-1} \frac{p_y}{p_x}$.

The impact of a 2D crystalline arrangement on the TCIF is best described in a graphical representation, as shown in Figure 1 (a modified version from Philippsen et. al 2007B). In the general case, at each frequency $\mathbf{p}$ in the Fourier representation of the measurement Q, information from two neighboring frequencies in the sample Φ are combined (Figure 1B). For an (idealized) 2D crystal, the Fourier domain consists of a set of discrete peaks, and each of those will show up twice in Eq.(1), as indicated in Figure 2C (for the equally valid case of a 1D crystal). The resulting peak splitting is a well-known phenomenon at high resolution reflections in experimental imaged of 2D crystals. As already eluded to in the first TCIF publication (Philippsen et al 2007B), this splitting is asymmetric, a feature which will be subject to discussion below.

Derivation

The path from the TCIF to the TTF formalism is easily shown with the graphical representation introduced above, as presented in Figure 2: The two frequencies of the split peaks p_1 and p_2 are approximated with the originating frequency q. As a consequence, the splitting becomes equidistant from q, and the phase shift introduced in the split peaks becomes symmetric. This in turn can be described as a local convolution of each original reflection in the sample with a Fourier transform of a sine wave, since

$$\begin{aligned} e^{-iW_0(q)}\delta\left(t - \tfrac{1}{2}q^2\lambda \tan\alpha\right) - e^{iW_0(q)}\delta\left(t + \tfrac{1}{2}q^2\lambda \tan\alpha\right) &= \mathrm{FT}\left[\sin\left(\tfrac{1}{2}q^2\lambda x \tan\alpha + W_0(q)\right)\right] \\ &= \mathrm{FT}\left[\sin(Ax + B)\right] \end{aligned} \tag{3}$$

giving

$$Q_q(t) = \mathrm{FT}\left[\sin(Ax + B)\right] \otimes \Phi_q(t) \tag{4}$$

where the subscript q denotes a local region around the reflection at that frequency. This corresponds to the TTF formalism introduced by Henderson et al (1990), yet derived differently.

An alternative path from the TCIF to the TTF can be attempted purely on equation terms. We have to say attempted because we have not been able to re-trace all derivation steps presented in the original TTF publication. In that publication, the derivation of the TTF starts with the basic concept of the CTF in frequency space (denoted by kernel H), which is multiplied with the Fourier transform of the sample (Φ) to yield the untilted measurement:

$$\begin{aligned} M(t) &= H(t)\,\Phi(t) \\ &= 2\sin(At^4 + B\,\Delta z\,t^2)\,\Phi(t) \end{aligned} \tag{5}$$

The description of the CTF contains the (constant) defocus Δz as a parameter (plus other parameters, simply given here as A and B). For the case of a tilted sample, however, the defocus is varying across the sample, e.g. for the linear case one may write $\Delta z(x) = \Delta z_0 + x\tan\alpha$, where z_0 is some base defocus at the origin, x is the spatial variable perpendicular to the tilt axis, and α is the tilt angle.

It is tempting to simply insert the defocus variation into the above formula, but this is problematic, since the spatial variable x is not directly applicable in the frequency domain. As a consequence, this problem can be circumvented by re-writing Eq.(5) as a convolution in real-space (which is equivalent to a multiplication in Fourier-space).

$$m(x) = \int d\xi \underbrace{\int dt\, e^{2\pi i t(x-\xi)}\, 2\sin(At^4 + B\,\Delta z\,t^2)}_{\mathrm{FT}^{-1}[H]}\,\phi(\xi) \tag{6}$$

As far as we can infer from the published TTF derivation, only part of this Eq.(6) is utilized, namely the sine and ϕ terms; furthermore, a spatial dependence is added to the sine term, by replacing Δz with the linear term $\Delta z(x)$ from above:

$$\begin{aligned} \hat{m}_t(\xi) &= 2\sin(At^4 + B\,\Delta z_0\,t^2 + B\,\xi\tan\alpha\,t^2)\,\phi(\xi) \\ &= 2\sin(C_t\xi + D_t)\,\phi(\xi) \\ &= \tilde{H}_t(\xi)\,\phi(\xi) \end{aligned} \tag{7}$$

We judge this step as problematic, since the encompassing integrals have been disregarded, and we do not understand the underlying mathematical justification.

Nevertheless, Eq.(7) forms the basis for the TTF: it's interpretation leads to the description of the tilt effect as a scattering angle (t) and spatial position (ξ) dependent contrast modulation of the sample ϕ, i.e. for each scattering angle, the actual sample appears to have a sine induced modulation in direction

perpendicular to the tilt axis.

At this point, Eq.(7) is seen to lie in the spatial domain, and the multiplication is converted into a convolution in a "new" Fourier domain, which is only valid for the frequency *t* (from the "old" Fourier domain):

$$\hat{M}_t(p) = \mathrm{FT}\left[\tilde{H}_t(\xi)\right] \otimes \Phi(p) \tag{8}$$

$\hat{M}_t(p)$ is taken to be the region around a reflection in the Fourier transform of the measurement.

We think that this step is also problematic, because it would not have been possible under the integrals in Eq.(6); re-casting formulas from the Fourier into the spatial domain should in our opinion not be done light-heartedly, especially in the case of tilted samples where a spatial dependence is present.

Sticking to the TTF interpretation though, the correction aims to "flip" the negative contrast of the apparent sample modulation by re-convoluting each $\hat{M}_t(p)$ with the corresponding $\mathrm{FT}\left[\tilde{H}_t(\xi)\right]$, which essentially translates to squaring the sine term in Eq.(7).

In fact, the following is a direct translation of the TTBOX code into the mathematical notation used here, where the region around each reflection peak ($\hat{M}_t$) is convoluted with the Fourier transform of $\tilde{H}_t$ to yield a TTF corrected reflection region Ξ_t.

$$\Xi_t(p) = \int dk \left[\int dx\, \tilde{H}_t(x) e^{-2\pi i x(p-k)} \right] \hat{M}_t(k) \tag{9}$$

This correction could actually make sense, if the previous steps would be a correct representation of the physical reality. As alluded to above, we are unsure whether this assumption holds.

Re-arranging Eq.(8) (akin to Eq.(9)) can shed some light on the main differences to the TCIF derivation:

$$\begin{aligned}
\hat{M}_t(p) &= \int dk \left[\int dx\, \tilde{H}_t(x) e^{-2\pi i x(p-k)} \right] \Phi_t(k) \\
&= \int dk \int dx\, \tilde{H}_t(x) e^{-2\pi i x p} e^{2\pi i x k} \Phi_t(k) \\
&= \int dx\, e^{-2\pi i x p} \tilde{H}_t(x) \int dk\, e^{2\pi i x k} \Phi_t(k) \\
&= \int dx\, e^{-2\pi i x p} \tilde{H}_t(x) \phi_t(x)
\end{aligned} \tag{10}$$

For the TCIF, we also have the $\tilde{H}$ term, in a seemingly similar expression (see Philippsen et al, 2007B):

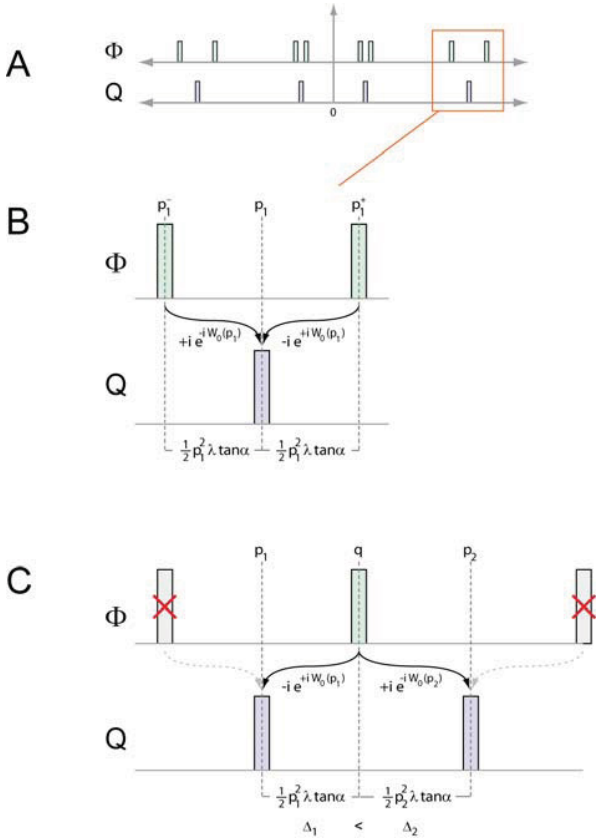
$$Q(p) = \int dx\, e^{-2\pi i x p} \tilde{H}(p,x)\, \phi(x) \tag{11}$$

The crucial difference in these two integral equations is the non-separable kernel $\tilde{H}$ in Eq.(11) as opposed to a trivial kernel in Eq.(10), which makes the TTF a convolution, but the TCIF not.

Discussion

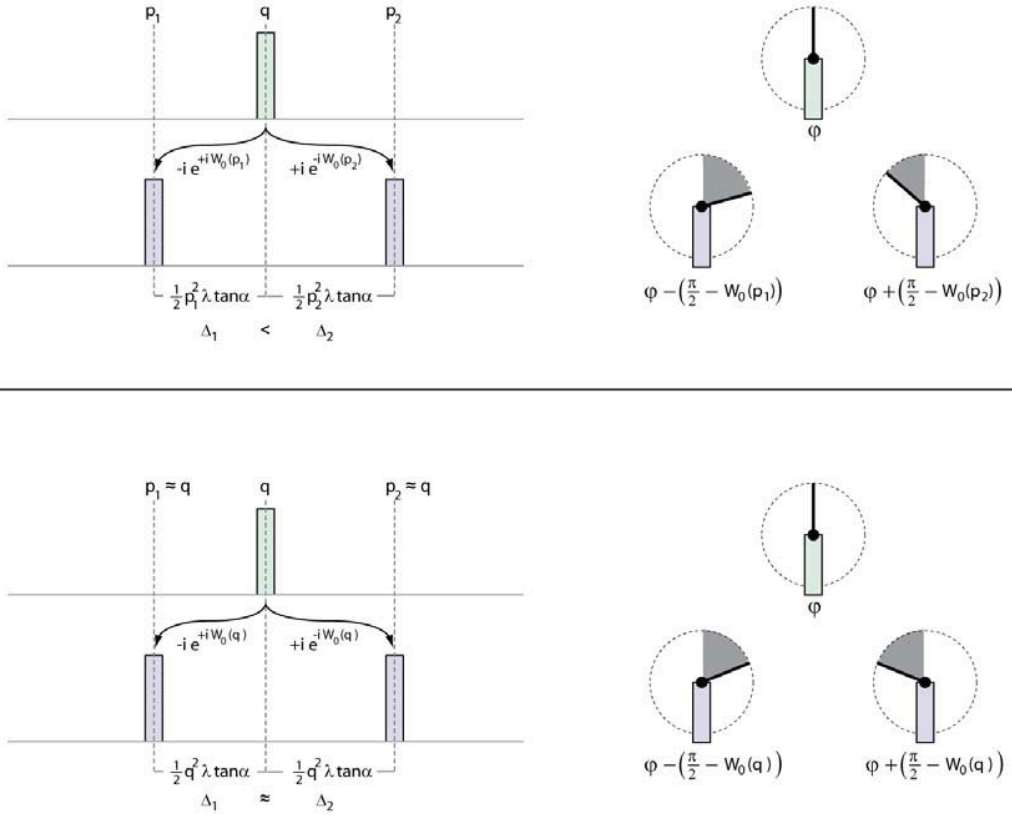
The above derivation shows that the TTF formalism is only an approximation of the full TCIF, and this would suggest that the TTF correction scheme as currently employed in the MRC suite may not be able to correctly remove the TCIF induced optical effects. This correction scheme has been used in the structural studies of several membrane proteins, however (Just to name a few examples: Murata et al. (2000), Mitsuoka et al. (1999), Holm et al. (2006) and Jegershoeld et al. (2008)), and thus we need to address the question under which conditions and up to which resolution the TTF correction can be safely applied.

Figure 1



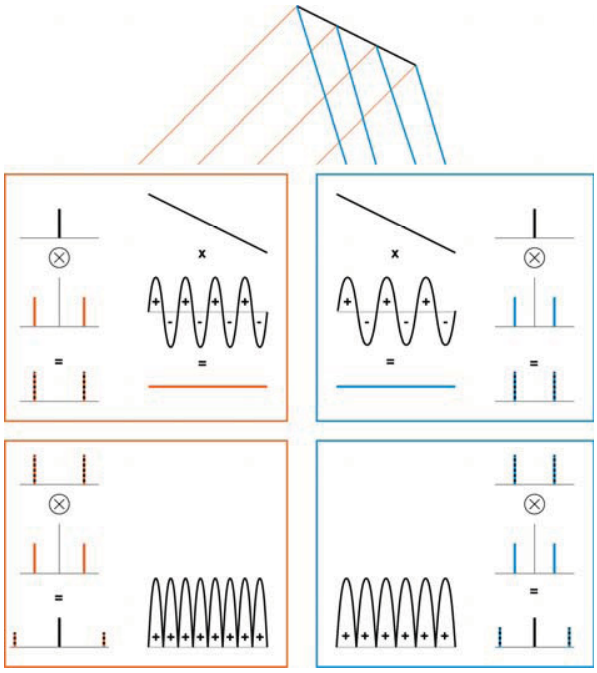
A graphical representation of TCIF effect in Fourier space, for the general case (**B**) and the crystalline one (**C**)

Figure 2



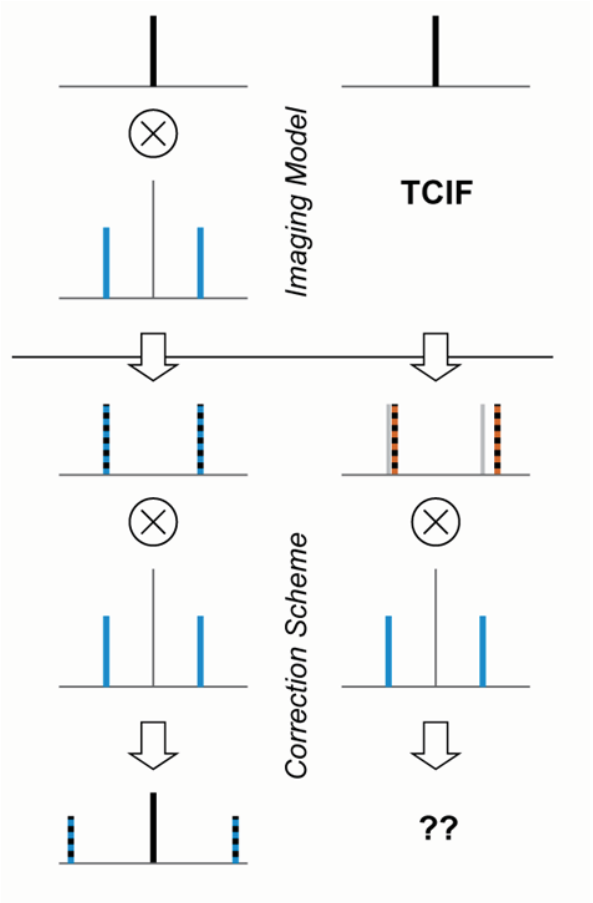
Graphical derivation of the TTF from the TCIF, based on the assumption that p1 and p2 can be replaced by q.

Figure 3



The consequence of the TTF imaging model, and the corresponding correction scheme. If the TTF were an accurate description of the imaging process, the correction would always work.

Figure 4



On the left the assumed combination of imaging and correction, as presented by Henderson et al (1990). On the right, the imaging step has been replaced by the TCIF, and the question remains what happens when this is subjected to the TTF correction - i.e., to what extent is the current application of the TTF correction algorithm in MRC valid.

Chapter 4

Annex - On The Correction of tilted 2D Crystals

In the "Derivation" section of Chapter 3 it has been shown how the formalism of the TTF can be derived from the one of the TCIF by introducing an approximation. In this chapter the consequences of this approximation will be discussed, especially focusing on their effects on the performance of the correction scheme proposed by Henderson et al. (1986) . The goal of this analysis is to determine if the correction strategy still allows an accurate recovery of the amplitude and phase information when the TTF approximation is dropped.

1 Peak Splitting Under the TTF approximation

According to the TTF model, the region of Fourier space surrounding a reflection is transformed by the imaging process in a way described by the following equations (See (7) and(8) in Chapter 3):

$$\widehat{M}_t(p) = FT[\tilde{H}_t(\xi)] \otimes \Phi(p) \tag{4.1}$$

$$\tilde{H}_t(\xi) = 2\sin(C_t + D_t) \tag{4.2}$$

In Chapter 3 it has been shown how this transformation is equivalent to the one applied by the TCIF model when the following approximation is considered valid:

$$\boldsymbol{p_1} = \boldsymbol{p}_2 = \boldsymbol{q} \tag{4.3}$$

As shown in Figure 2 in Chapter 3, this approximation makes the peak splitting process symmetric. The frequency separation of both split peaks from the original reflection becomes identical:

$$|p_2 - q| = |p_1 - q| = \frac{1}{2}\boldsymbol{d}p^2\lambda\tan\alpha \tag{4.4}$$

and the phase separation that each split peak has from the original reflection becomes symmetric:

$$P_1 = -iAe^{i\varphi}e^{iW_0(q))} \tag{4.5}$$

$$P_2 = iAe^{i\varphi}e^{-iW_0(q))} \tag{4.6}$$

(In this expressions, P_1 and P_2 are the complex values of the two split peaks, while A and φ are respectively the amplitude and the phase of the original reflection).

When the approximation is not valid, the system stays asymmetric. Figure 1 in Chapter 3 shows that the spatial frequencies of the two split peaks must then satisfy the conditions:

$$p_1 + \frac{1}{2}p_1^2\lambda\tan\alpha = q \tag{4.7}$$

$$p_2 - \frac{1}{2}p_2^2\lambda\tan\alpha = q \tag{4.8}$$

Solving the two equations according to p_1 and p_2 gives:

$$p_1 = \frac{-1 + \sqrt{1 + 2q\lambda\tan\alpha}}{\lambda\tan\alpha} \tag{4.9}$$

$$p_2 = \frac{1 - \sqrt{1 - 2q\lambda\tan\alpha}}{\lambda\tan\alpha} \tag{4.10}$$

p_1 and p_2 are not symmetric around q.

The phase shifts are also not symmetric. The complex values of the split peaks are related to the value of the original Fourier element by the following relations:

$$P_1 = -iAe^{i\varphi}e^{iW_0(p_1)} \tag{4.11}$$

$$P_2 = iAe^{i\varphi}e^{-iW_0(p_2)} \tag{4.12}$$

The phase shifts have opposite sign, but not the same magnitude, as the W_0 terms in the equations depend on the local spatial frequencies of the split peaks, and not on the frequency of the original reflection.

Table 4.1 summarizes the characteristics of the split peaks when the TTF approximation is held valid and when it is dropped. It is worthwhile to note that under the TCIF model, when a reflection lies at low spatial frequency, the separation of the split peaks is small, as is the difference in the magnitude of the two phase shifts. The TCIF model is, under this conditions, almost symmetric, and the TTF approximation can be considered valid. However, as the spatial frequency of the reflection increases, the system becomes increasingly asymmetric, and the approximation loses its validity.

The importance of the symmetric nature of the system under the TTF approximation cannot be overestimated. It will be shown in the next paragraphs that the TTF-based correction method relies on it to work.

	With TTF approximation	Without TTF approximation
Left peak frequency (p_1)	$\mathbf{p}$-$\frac{1}{2}p_1^2\lambda\tan\alpha$	$\frac{-1+\sqrt{1+2q\lambda\tan\alpha}}{\lambda\tan\alpha}$
Left peak amplitude	A	A
Left peak phase	$\varphi-\frac{\pi}{2}+W_0(q)$	$\varphi-\frac{\pi}{2}+W_0(p_1)$
Right peak frequency (p_2)	$\mathbf{p}$+$\frac{1}{2}p_1^2\lambda\tan\alpha$	$\frac{1-\sqrt{1-2q\lambda\tan\alpha}}{\lambda\tan\alpha}$
Right peak amplitude	A	A
Right peak phase	$\varphi+\frac{\pi}{2}-W_0(q)$	$\varphi+\frac{\pi}{2}-W_0(p_2)$

Table 4.1 Summary of the characteristics of the split peaks. The frequency of the original reflection is q, its amplitude is A and its phase is φ.

2 The TTF Correction Method

The nature of the TTF correction method has been expressed in Chapter 3 of this manuscript in the following form (See equation (9) in Chapter 3):

$$\Xi_t = \int dk \left[\int dx \tilde{H}_t(\xi) e^{-2\pi i x(p-k)} \right] \widehat{M}_t(k) \tag{4.13}$$

This amounts basically to a convolution of the area of Fourier space around the reflection (which contains the split peaks) with the Fourier transform of $\tilde{H}_t(\xi)$

$$\Xi_t = FT[\tilde{H}_t(\xi)] \otimes \widehat{M}_t(p) \tag{4.14}$$

Let us now consider a reflection lying at spatial frequency q in the Fourier transform of a 1-dimensional image. The conclusions that will be drawn can easily be expanded to 2 dimensions.

$\tilde{H}_t(\xi)$ is a sin function. Its Fourier transform has the form of a δ function in Fourier space multiplied by a complex number, accompanied by its conjugate companion:

$$FT[\tilde{H}_t(\xi)] = 2\delta(\frac{1}{2}q^2\lambda\tan\alpha)e^{i\left(\frac{\pi}{2}-W_0(q)\right)} + 2\delta(-\frac{1}{2}q^2\lambda\tan\alpha)e^{i\left(-\frac{\pi}{2}+W_0(q)\right)} \tag{4.15}$$

(In this expression, the term C_t and D_t have been expanded according to equation (3) in Chapter 3).

Two characteristics of this transform are important when discussing the nature of TTF correction method. The first is the frequency separation between the two peaks. The second is the phase value of complex number that multiplies each δ function.

Peak Separation in the Convolution Kernel

In Henderson et al. (1986) , the author says about the TTF correction method that "The result is to create a corrected transform with a diffraction peak in

the correct position". This effect stems directly from the fact that under the TTF approximation, the separation of the two δ function peaks in the Fourier transform of $\tilde{H}_t(\xi)$ matches the distance between the split peaks generated by the imaging process.

$$\Xi_t(q) = 2P_l e^{i\left(\frac{\pi}{2} - W_0(q)\right)} + 2P_r e^{i\left(-\frac{\pi}{2} + W_0(q)\right)} \tag{4.16}$$

and analysis of this combined peak allows the recovery of the information of the original reflection. Equation 4.16 can be written as:

$$\Xi_t(q) = 2Ae^{i\left(\varphi - \frac{\pi}{2} + W_0(q)\right)} e^{i\left(\frac{\pi}{2} - W_0(q)\right)} + 2Ae^{i\left(\varphi + \frac{\pi}{2} - W_0(q)) + \gamma_1\right)} e^{i\left(-\frac{\pi}{2} + W_0(q)\right)} \tag{4.17}$$

Which gives:

$$\Xi_t(q) = 2Ae^{i\varphi} + 2Ae^{i\varphi} = 4Ae^{i\varphi} \tag{4.18}$$

The phases of the peak and of the original reflection are identical, while the amplitudes are proportional. The nature of this proportionality is strongly tied to the phase shifts of the δ peaks in the convolution kernel, as will be shown in the next paragraphs.

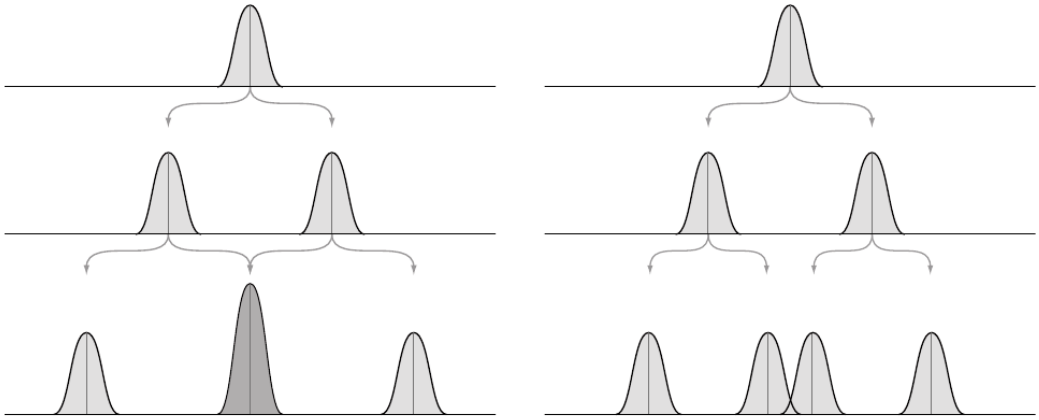


Figure 4.1 Effects of the convolution of the split peaks generated during the imaging process with the Fourier transform of $\tilde{H}_t(\xi)$. The image on the left shows the effects of the convolution when the separation of the peaks in the convolution kernel matches the one of the sub split peaks. Both the sub peaks project information at the spatial frequency of the original reflection q. The information is used to recover phase and amplitude of the original element. The image on the right shows the effects of the convolution when the separation of the two pairs of peaks is not the same: the information from the split peaks is not projected to the same spatial frequency. Two distinct peaks, each carrying some information from the original reflection, are formed near q, but the information they convey is not combined. In experimental images, peaks are not δ function-shaped, but broad as shown in the image. This can lead to interaction between non-combined peaks (image on the right) with recovery of correct phase information.

Phase Shifts in the Convolution Kernel

Equation 4.15 shows that the two δ peaks in the Fourier transform of $\tilde{H}_t(\xi)$ have phases of $\frac{\pi}{2} - W_0(q)$ and $-\frac{\pi}{2} + W_0(q)$. From table 4.1, it can be seen that under the TTF approximation, these are the same phase shifts that the split peaks have with respect to the phase of the original reflection.

This condition is not a requirement to recover the phase information of the original reflection, but it can be shown that it must be satisfied to allow the recovery of the amplitudes of all the reflections with the correct relative magnitude.

Let the the Fourier transform of $\tilde{H}_t(\xi)$ have the following form:

$$FT[\tilde{H}_t(\xi)] = 2\delta(\frac{c}{2\pi})e^{-iW_1} + 2\delta(-\frac{c}{2\pi})e^{iW_1} \tag{4.19}$$

The two delta peaks still show symmetric phase shifts, but with a magnitude that is different from the phase shifts the two split peaks.

The convolution will then bring the following complex value at the spatial frequency of the original reflection:

$$\Xi_t(q) = 2Ae^{i\left(\varphi-\frac{\pi}{2}+W_0(q)-W_1\right)} + 2Ae^{i\left(\varphi+\frac{\pi}{2}-W_0(q)+W_1\right)} \tag{4.20}$$

This can be written as:

$$\Xi_t(q) = [e^{i\left(-\frac{\pi}{2}+W_{0(q)}-W_1\right)} + e^{-i\left(-\frac{\pi}{2}+W_0(q)-W_1\right)}]2Ae^{i\varphi} \tag{4.21}$$

Expressing the sum of exponentials in trigonometric form:

$$\Xi_t(q) = [\cos\left(\tau\right) + i\sin(\tau) + \cos(-\tau) + i\sin(-\tau)]2Ae^{i\varphi} \tag{4.22}$$

$$\tau = \left(-\frac{\pi}{2} + W_0(p) - W_1\right) \tag{4.23}$$

Since $\sin(-\tau) = -\sin(\tau)$ and $\cos(-\tau) = \cos(\tau)$, the previous expression becomes:

$$\Xi_t(q) = [2\cos(\tau))]2Ae^{i\varphi} \tag{4.24}$$

The combined peak and the original reflection still have the same phase. The amplitudes are also proportional, but the proportionality coefficient depends on the values of the phases shifts, which depend, in turn, on the frequency of the original reflection. The amplitude of each reflection is recovered with a different coefficient, and information concerning the relative magnitude is lost completely. In particular for some reflections the coefficient can be negative, leading to the assignment of the wrong sign to the phase, or even null, rendering the recovery completely impossible. The coefficient value becomes independent from the phase shifts and assumes the value of 4 only when $\tau = 0$, which leads to:

$$W_1 = -\frac{\pi}{2} + W_0(p) \tag{4.25}$$

This proves that only when the phase shifts of the split peaks and of the delta functions in the convolution kernel are identical, the relative magnitude information is preserved. Under the TTF approximation, this condition is satisfied.

It is interesting to note that what allows the recovery of the correct reflection phase information even when the phase shifts do not match is the symmetry of the system. The convolution process combines the components projected by the two split peaks in a single complex value. Being symmetric, and symmetrically combined, the two phase shifts cancel each other, recovering the phase of the original reflection. Obviously, this canceling effect does not preserve the amplitude information. For its correct recovery, the effects of the phase shifts need to be reversed completely during the convolution, as described above.

3 Dropping the TTF Approximation

In the previous paragraphs it was shown that the TTF approximation can be considered valid only at low spatial frequencies, where the separation of the split peaks is small. As the spatial frequency of the processed reflection increases the approximation needs to be dropped.

The main effect is that the split peak system becomes asymmetric.In particular, the separation and the phase shifts of the δ peaks in the Fourier transform of $\tilde{H}_t(\xi)$ stop matching the ones of the split peaks. The TTF correction method should then stop working. Several high resolution structures have however been published using the TTF method to correct the effects of the tilt (for example Murata et al. (2000) and Mitsuoka et al. (1999) , Ren et al. (2000) or Jegershoeld et al. (2008) just to name a few). Obviously, the method still retains some correction power even when the TTF approximation cannot be considered valid. The following paragraphs will discuss how this is possible.

Consequences of the Asymmetry of the Peak Splitting

The plot in figure 4.2 represents graphically the mismatch between the peaks of the split Fourier element and the delta function peaks in the convolution kernel for typical electron crystallography imaging conditions. The distance in frequency units between the two peaks is shown for Fourier elements lying in a wide range of frequencies. The line in the top part of the plot refers to the peaks lying at lower spatial frequency that the original Fourier element, while the line in the bottom half refers to the peaks lying at higher spatial frequency.

It can be seen that the mismatch is very limited, of the order of a fraction of a frequency unit, up until very high spatial frequencies. The actual data processing is done on images, which are discretized datasets: minutes differences can

be ignored when they are much smaller smaller that the discretization sampling step. This allows the TTF correction method to be effective up to almost atomic resolution. Furthermore, peaks in real experimental dataset are not sharp delta functions. The crystalline structure being imaged is not perfect and the extent of the image is not infinite. This causes a broadening of the peaks which lose their δ function shape. Even when the TTF correction does not project all the information in a single peak but results in two separate peaks, their broad tails might overlap and combine at the spatial frequency of the original reflection (see Figure 4.1). In real experiments, this effect can be overpowered by a instrumental noise level. It can lead, nevertheless, to a slight increase of the frequency range in which the TTF correction method still can work. Because of all these considerations, it will be assumed, in the discussion in the next paragraphs, that TTF correction can project some information at the frequency of the original reflection at all but the highest resolutions.

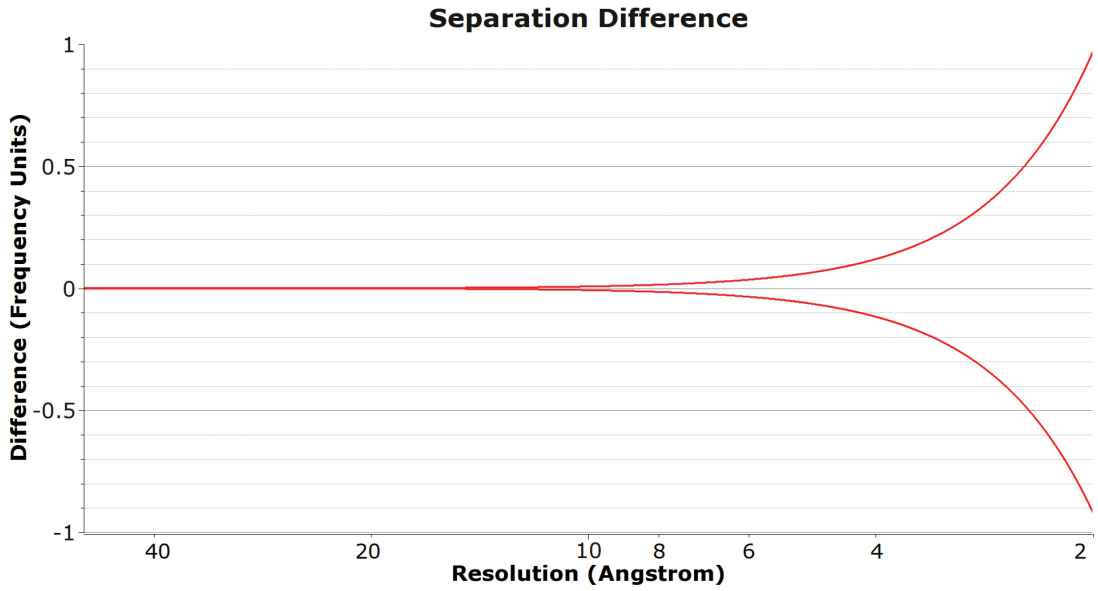


Figure 4.2 Mismatch between the peaks of the split Fourier element (according to the TCIF model) and the delta function peaks in the TTF convolution kernel. The plot represents the mismatch in frequency units under typical electron crystallography imaging conditions (Image size: 8000x8000 pixels, pixel width: 1.0 Angstrom, Defocus: 1 micrometer, acceleration voltage 200kV). The line in the top part of the plot refers to the peaks lying at lower spatial frequency that the original Fourier element, while the line in the bottom half refers to the peaks lying at higher spatial frequency. The mismatch becomes relevant only at spatial frequencies close to atomic resolution.

Consequences of the Asymmetry of the Peak Phase Shifts

Figure 4.3 shows the difference in degrees between the phase shifts of the split peaks and the ones of the convolution kernel under typical electron crystallography imaging conditions, and for different spatial resolutions. Once again, the line in the top part of the plot refers to the peaks lying at lower spatial frequency than the original Fourier element, the one in the bottom half of the plot refers to the peaks lying at higher spatial frequency. It can be seen that the difference starts to be significant at average spatial frequencies, far below atomic resolution. Close to atomic resolution, the asymmetry of the difference becomes also evident.

These conditions lead the TTF correction to project at the frequency of the original Fourier element the following information:

$$\Xi_t(q) = 2Ae^{i\left(\varphi-\frac{\pi}{2}+W_0(p_1)\right)}e^{i\left(\frac{\pi}{2}-W_0(q)\right)} + 2Ae^{i\left(\varphi+\frac{\pi}{2}-W_0(p_2)\right)}e^{i\left(-\frac{\pi}{2}+W_0(q)\right)} \tag{4.26}$$

This can be written as:

$$\Xi_t(q) = 2Ae^{i(\varphi+W_0(p_1)-W_0(q))} + 2Ae^{i(\varphi-W_0(p_2)+W_0(q))} \tag{4.27}$$

By collecting the common terms:

$$\Xi_t(q) = [e^{i(W_0(p_1)-W_0(q))} + e^{i(-W_0(p_2)+W_0(q))}]2Ae^{i\varphi} \tag{4.28}$$

The presence of the summation term in equation 4.28 is extremely important, as it introduces errors in the determination of both phase and amplitude of the reflection. The nature of these errors deserves extensive discussion.

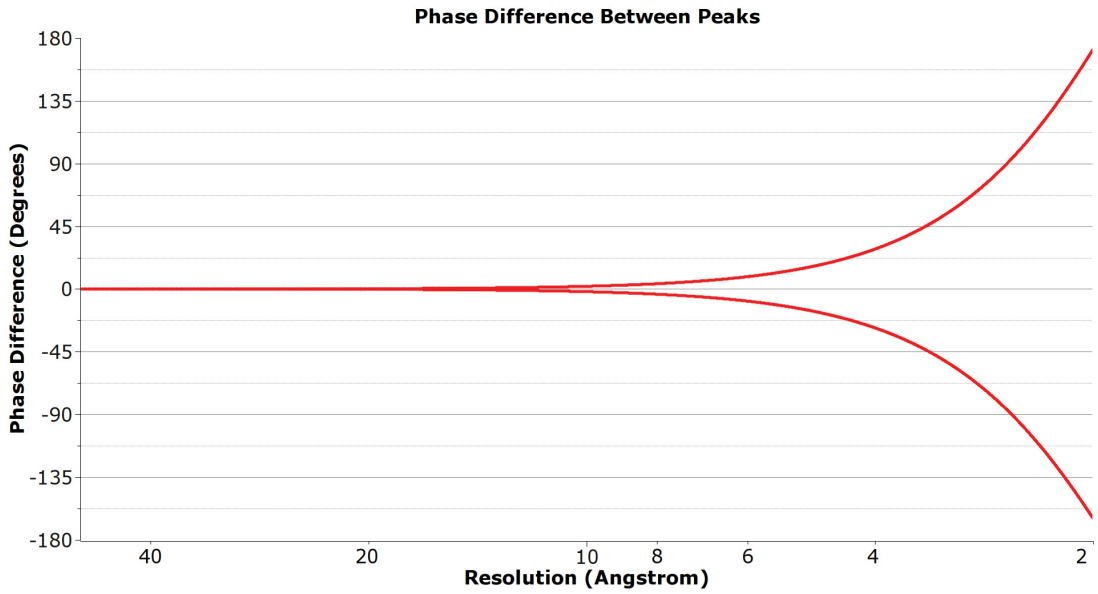


Figure 4.3 Difference in degrees between the phase shifts of the peaks of the split Fourier element (according to the TCIF model) and the ones of the TTFconvolution kernel. The plot represents the difference under typical electron crystallography imaging conditions (Image size: 8000x8000 pixels, pixel width: 1.0 Angstrom, Defocus: 1 micrometer, acceleration voltage 200kV). The line in the top part of the plot refers to the peaks lying at lower spatial frequency than the original Fourier element, the one in the bottom half of the plot refers to the peaks lying at higher spatial frequency. The difference starts to become significant at intermediate resolution, although the asymmetry of the difference itself starts to show only at spatial frequencies corresponding to atomic resolution.

Error in the Determination of the Phase

At low spatial frequencies, where the TTF approximation holds:

$$p_l \backsimeq p_x \simeq p_r \tag{4.29}$$

And:

$$W_0\left(p_l\right) \simeq W_0\left(p_x\right) \simeq W_0\left(p_r\right) \tag{4.30}$$

The combined information has the form:

$$Q_c(p_x) = 4Ae^{i\varphi} \tag{4.31}$$

Allowing an accurate recovery of the phase. However, as the approximation loses validity, the summation term in equation 4.28 starts to assume a non-null value, and to introduce an error in the recovery of the phase information.

This term consists in the sum of two complex numbers with same module but different phase. Since:

$$W_0(p_1) < W_0(q) < W_0(p_2) \tag{4.32}$$

their phases have the same sign and don't cancel each other. Furthermore, since the difference between p_1 and q and p_2 and q increases as q increases, the error becomes bigger as the spatial frequency of the reflection increases.

The presence of the summation term has also an impact on the recovery of the amplitude information. The analysis of the combined peak still returns an amplitude value which is proportional to the one of the reflection, but information about relative amplitude of a number of different reflections is lost.

Loss of Relative Amplitude Information

The summation term combines two complex elements and can be represented graphically in the Argand plane as the sum of vectors shown in Figure 4.4. It is obvious from simple trigonometric considerations that the modulus of the sum vector depends on the relative angle of the vectors being summed. This causes the modulus of the summation term to vary depending on the exponents of the two elements, which in turn depend on the spatial frequency of the reflection being processed.

The complex number projected at frequency q still has a modulus which is proportional to the one of the original reflection, but the proportionality coefficient depends on q. The amplitude of each reflection is then recovered with a different coefficient, and information regarding the relative amplitudes is lost. Although the structural information is carried mostly by the phases, the relative amplitudes are nevertheless important, especially for intermediate processing steps like the merging of the data from different experiments or the computation of the so called "lattice lines" along the reciprocal z axis.

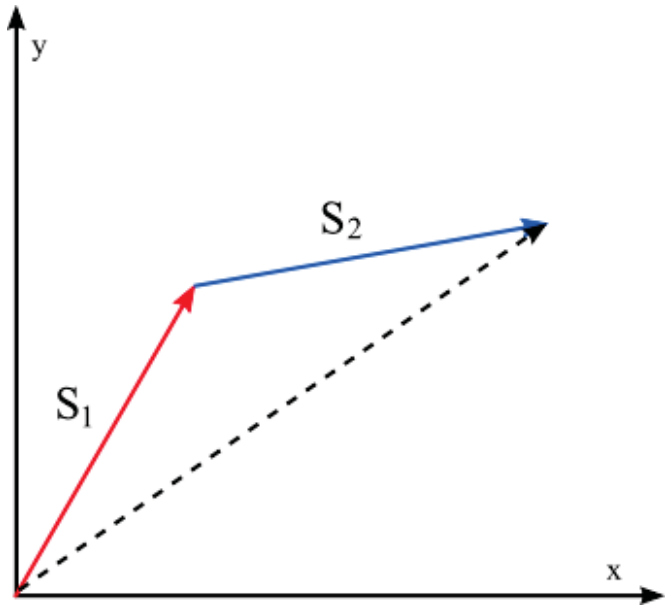


Figure 4.4 The summation term in equation 4.28 can be represented graphically in the Argand plane as the sum two vectors S_1 and S_2 with the same amplitude and different phases. The phase of the sum vector represents the error with which the phase information is recovered by the correction method when the TTF approximation is dropped. It is obvious that the modulus of the sum vector depends on the relative angle (phases) of vectors S_1 and S_2, and this causes the proportionality coefficient with which the amplitude information is recovered to be dependent on the frequency of the processed reflection.

Estimation of the Errors through Simulation

In order to estimate the overall magnitude of all the the effects described in the previous paragraphs, a simulated correction process was implemented using synthetic images. The results are summarized in figure 4.5 and figure 4.6, which show the typical error involved in the recovery of the respectively the phase and the ampplitude information for reflections lying at different resolutions, under typical electron crystallography imaging conditions.

The plots in the figures have been generated using simulated images of crystalline samples. A simulated 1-dimensional image of size 8000x1 pixels was created. The periodicity of the crystal structure was simulated by placing reflection peaks at a distance of 80 frequency units in the Fourier transform of the image. All the peaks had an amplitude value of 255 and a phase of 0. The effects of the tilt were then applied using the TCIF model, assuming a tilt angle of 60 degrees and a tilt axis lying perpendicular to the extent of the image. The TTF correction was finally performed. The recovered amplitudes and the phases were compared with the original ones. Figure 4.5shows the absolute error (in degrees) in the recovery of the phase information for a tilt angle of 60 degrees . Figure 4.6 shows instead the ratio between the recovered amplitude and the amplitude of the original reflections at the same tilt angle.

The error in phase recovery increases steadily with resolution. Although the absolute magnitude of this error is limited at resolutions other than atomic, the error is nevertheless present and influences all the image processing steps that follow the TTF correction, combining with the errors introduced by all the other image processing steps. Furthermore, at atomic resolution the error becomes so big that the recovered phase has essentially a random value. Eventually, this can lead to a lowering of the resolution of the final computed structure.

The absolute ratio between the recovered amplitude and the original one depends on the rescaling steps performed during the image processing. Its is clear, however, that the ratio should be the same for all reflections for relative amplitude information to be preserved. This clearly does not happen, even at moderate resolution. At atomic resolution, the difference becomes dramatic. The TTF correction method does not preserve very well relative amplitude information.

It should also be pointed out that the analysis that was carried out in this section of the manuscript assumed ideal conditions: perfect crystalline structures, infinitely sharp peaks and no instrumental noise. As a consequence the error estimations given in this chapter must be considered as a measure of the maximum accuracy limit that the algorithm allows under perfect conditions, not as assessment of the reliability of the method in real life experiments.

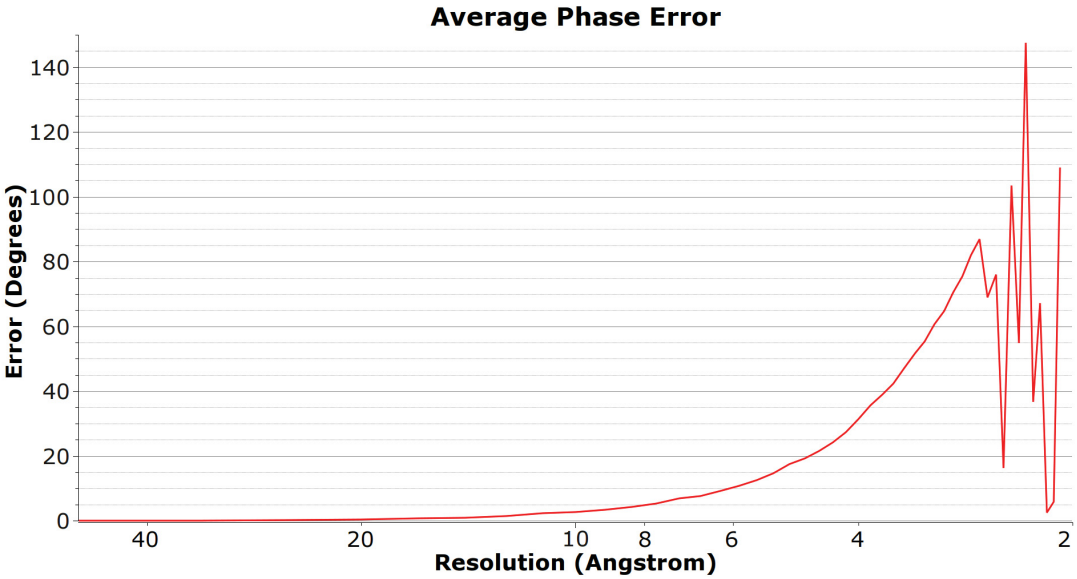


Figure 4.5 Error in the recovery of the phase information using the TTF correction with a tilt angle of 60 degrees (Under typical electron microscopy conditions). The plot shows the absolute difference in degrees between the recovered phase and the phase of the original reflection over a wide range of resolutions. The error increases steadily with the spatial frequency of the reflection and becomes significant in the vicinity of atomic resolution (Image size: 8000x8000 pixels, pixel width: 1.0 Angstrom, Defocus: 1 micrometer, acceleration voltage 200kV, distance of peaks in reciprocal space: 80 pixels).

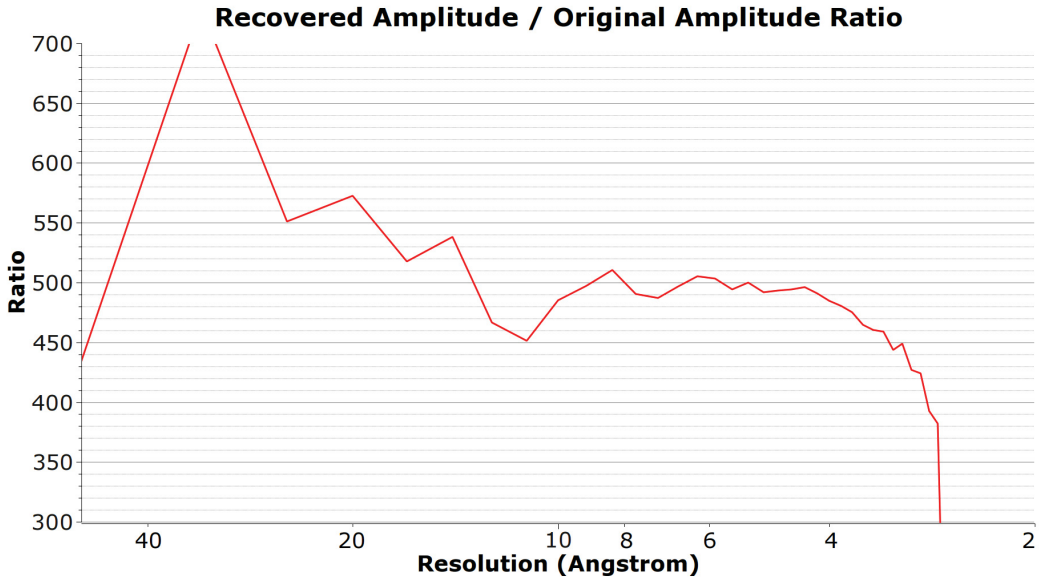


Figure 4.6 Loss of relative amplitude information when using the TTF correction with a tilt angle of 60 degrees (Under typical electron microscopy conditions). The plot shows the ratio between the recovered amplitude and the amplitude of the original reflection over a wide range of resolutions. The absolute value of the ration depends on rescaling operations performed by the image processing software. For relative amplitude information to be preserved, however, the ratio should be the same for all reflections. It can be seen that already at moderate resolution the ratio starts to differ from reflection to reflection, and the difference becomes dramatic as atomic resolution is approached.(Image size: 8000x8000 pixels, pixel width: 1.0 Angstrom, Defocus: 1 micrometer, acceleration voltage 200kV, distance of peaks in reciprocal space: 80 pixels).

4 Outlook and Future Developments

In this section of the manuscript, it has been shown that the TTF correction retains some correction power even when the TTF approximation is dropped. In recent years, several electron crystallography structures have been published where quasi-atomic resolution has been achieved(For example, Murata et al. (2000) (3.8 Angstrom) or Holm et al. (2006) (3.2 Angstrom), just to cite a few). This is in line with the analysis presented here: the limitations discussed in the previous paragraphs become significant only close to atomic resolution.

On the other hand these need to be addressed in order to put atomic resolution routinely within reach electron crystallography. It is interesting to note that equation 4.28 allows an estimation of the information recovery errors as the value of the summation term can be computed for reflections lying at different spatial frequencies. Due to the exponents of the two elements having the same sign, the error always has the same direction, a correction could be theoretically applied to the recovered phase information to correct it The same idea could obviously be applied to the recovery of the relative amplitude information as the summation term determines the proportionality coefficient for each reflection. It could be straightforward to scale each recovered amplitude value accordingly.

The real problem of the TTF correction method is that it relies on the presence of more or less sharp split peaks, whose information is combined by the convolution. In order to have sharp peaks, the computational correction of lattice defects must be performed before the effects of the imaging function are dealt with. It has been shown, however, that the correction of lattice defects ("unbending") leads to the deterioration of the local information carried by the Point Spread Function in real space (Schenk et al., 2009), and this leads to a corruption of the data in Fourier space. There is obviously no way out of this situation, and it is likely that new and different methods to correct the effects of

the tilt, like the stripe-based correction scheme described in Chapter 1 or the full inversion of the TCIF model described introduced in Philippsen et al. (2006) , will need to be used in the future.

5 Technical Note: MRC and IPLT

The TTMASK and TTBOX programs in the MRC package (Crowter et al., 1996) perform the correction of the tilt effect on images of crystalline samples using the TTF method discussed in the previous sections. TTMASK performs the correction on an image and returns it to the user to extract the amplitude and phase information. TTBOX already performs this task internally, returning to the user a list of diffraction peaks with associated amplitude and phase values.

All the simulations presented in this chapter were run using a reimplementation of the TTF correction algorithm within the IPLT development framework (Philippsen et al., 2007). This was done to achieve several goals:

1) To take advantage of the modern object-oriented C++ development framework to have a cleaner code, easier to debug than the old MRC FORTRAN code.

2) To avoid confusion between different conventions and physical units used by the program that simulates the tilt (IPLT) and the program that corrects it (MRC).

3) To obtain a clean implementation of the TTF algorithm in order to evaluate its advantages and limitations, taking out of the picture any additional heuristic correction that might have been introduced in MRC.

The IPLT implementation of the algorithms generates the convolution kernel directly in Fourier space, following the derivation presented in this chapter. The MRC implementation follows instead the classical derivation described in Henderson et al. (1986) and Henderson et al. (1990) : the $\widehat{H}_t$ function (called TTF function in the TTF model) is generated in real space, Fourier transformed , and convoluted with M_t.

The section of the TTBOX and TTMASK programs performing the correction is a subroutine called TTCORRECT.

TTCORRECT calls the CTFGEN function to compute the real space TTF function. CTFGEN uses two precomputed cofficients: $C1$ and $C2$. The coefficients have the following form (θ is the scattering angle, tied to the spatial frequency of the reflection q by the expression: $\theta = q\lambda$):

$$C1 = \frac{2\pi\theta^2}{2\lambda} \tag{4.33}$$

$$C2 = -C1Cs\frac{\theta^2}{2} \tag{4.34}$$

CTFGEN combines the two in the following way:

$$CHI = C1DF + C2 \tag{4.35}$$

$$DF = z_0 + (-x_p \sin\phi + y_p \sin\phi)\tan\alpha = z_0 + r\tan\alpha \tag{4.36}$$

Where x_p and y_p are spatial coordinates with respect to the center of the image, where the tilt axis is assumed to lie, and ϕ is angle between the tilt axis and the x axis of the reference system.

CHI is finally used to compute the TTF function, which is stored in the variable CNTRST.

$$CNTRST = -\sin(CHI) \tag{4.37}$$

$$CNTRST = -\sin[\frac{2\pi\theta^2}{2\lambda}(z_0 + \xi\tan\alpha) - \frac{2\pi\theta^2}{2\lambda}Cs\left(\frac{\theta^2}{2}\right)\Big] \tag{4.38}$$

$$CNTRST = -\sin\left[\xi\frac{\pi\theta^2}{\lambda}\tan\alpha + \left(\frac{\pi\theta^2}{\lambda}z_0 - \frac{\pi\theta^4}{2\lambda}C_s\right)\right] \tag{4.39}$$

Where:

$$\xi = (x_p, y_p) \tag{4.40}$$

This is equivalent to:

$$CNTRST = -\sin\left[\xi\pi\lambda q^2 \tan\alpha + W_0(q))\right] \tag{4.41}$$

The computed TTF function is then Fourier transformed by the TFXFFT function , and stored in the arrays ACTF and BCTF (the real and imaginary part respectively.) The TTCORRECT routine finally calls the CONVOLUTE function to perform the convolution.

The Fourier transform of equation 4.41 has the form of two δ functions multiplied with complex numbers. The δ peaks lie at the following spatial frequencies:

$$p_l = \frac{1}{2}q^2\lambda\tan\alpha \tag{4.42}$$

$$p_r = -\frac{1}{2}q^2\lambda\tan\alpha \tag{4.43}$$

They have unit amplitude of 2 and phases of:

$$\varphi(P_l) = \frac{\pi}{2} - W_0(q) \tag{4.44}$$

$$\varphi(P_r) = -\frac{\pi}{2} + W_0(q) \tag{4.45}$$

When the different sign conventions used for the CTF function in IPLT and MRC are taken into account, this convolution kernel appears identical to the one generated by IPLT (see table Table 4.1). This proves that even if different strategies are followed, the two versions of the algorithm behave essentially in the same way.

Chapter 5

Other algorithms

Until this chapter, the subject of this manuscript has been the analysis and simulation of the imaging process in tilted geometry. Here, however, some algorithms that were created outside that main body of work will be introduced. Several of the algorithms presented here were developed to facilitate the proof-of-concept implementation of an internal docking procedure in IPLT. Other algorithms are at the core of important steps in the IPLT processing pipeline for electron crystallography data. Still others were designed as stand-alone utilities. For each algorithm, some background is given, and the reasons that led to its development are detailed. This is followed by an analysis of the algorithm's nature and characteristics. Finally, the results that the use of the algorithm allowed to achieve and the issues that it raised are briefly discussed. In the section describing the implementation of each algorithm, the path that allows the recovery of the source code from the IPLT "site_alg" or "iplt" repositories is given.

1 LinearFit: Linear Fitting with Errors in X and Y

Introduction and Background

Crystalline samples in electron crystallography are very sensitive to radiation damage (Dorset, 1995). It is usually possible to collect only a single image or diffraction pattern before the specimen deteriorates to the point of being unusable. Hundreds of micrographs are however necessary to produce a meaningful 3-dimensional structure of the sample under investigation. Hence, electron

crystallography datasets come from homogeneous sources: each micrograph is collected from a different sample during a different experimental session. Ideally, the samples should be identical and experimental conditions should be reproduced from one session to the next. In practice, this rarely happens. When diffraction patterns and images in a dataset are processed, the amplitudes of the peaks in Fourier space appear to lie on different intensity scales. It is then necessary to let the data undergo a merging process, bringing all the elements on the same scale.

This merging process is very complicated and a detailed description of the its nature lies beyond the scope of this manuscript (Crowter et al., 1996). Suffices to say that it is based on a linear fitting procedure, and uses two parameters from each diffraction peak: $z*$, the frequency coordinate of the peak in the direction of the reciprocal z axis, and I, its intensity. Both parameters are experimental and come with uncertainty, and a linear fitting algorithm that can manage errors in both x and y coordinates is required to process them. This algorithm was developed to satisfy this need.

Implementation

(<iplt>/src/ex/alg/lib/linfit.hh)

The linfit algorithm contains a routine that performs a linear fit in absence of an error for the x coordinate, and another that can fit data with errors in both coordinates. Depending on the type of data being processed, the correct routine is called. The part of the algorithm that manages errors in both coordinates follows closely the program *fitexy* described in the book "Numerical recipes in C++" (Vetterling and Flannery, 2002) (Chapter 15, section 3, "Straight-Line Data with Errors in both coordinates", p. 671). The routine works by minimizing a scoring function of the form:

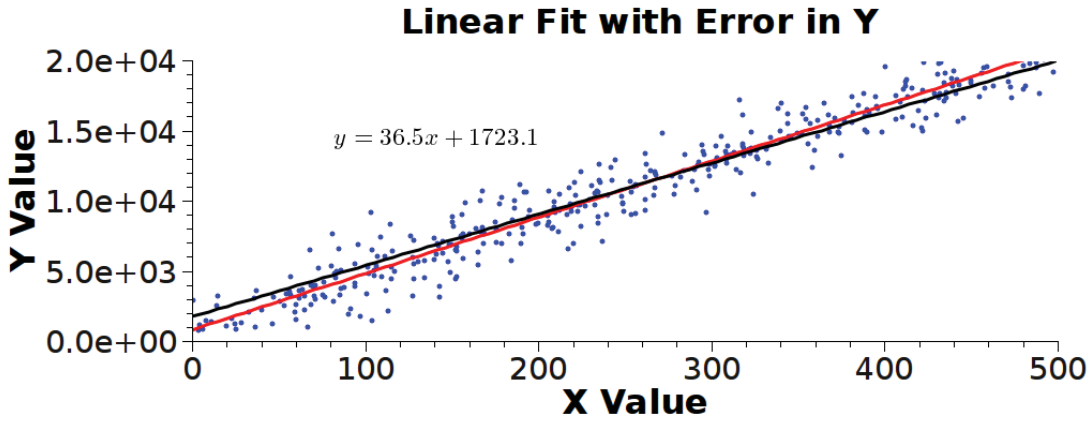
$$\chi^2(a,b) = \sum_{i=0}^{N-1} \frac{(y_i - a - bx_i)^2}{\sigma_{yi}^2 + b^2\sigma_{xi}^2}$$

using an iterative procedure. In this analytical expression, x_i and $y_i$are the coordinates of the data item i, and σ_{xi} and σ_{yi} are the associated errors. The program code presented in the book has been cleaned up and updated to a C++ object-oriented style of programming. Helper methods and functions have been added to facilitate the handling of the input and output data.

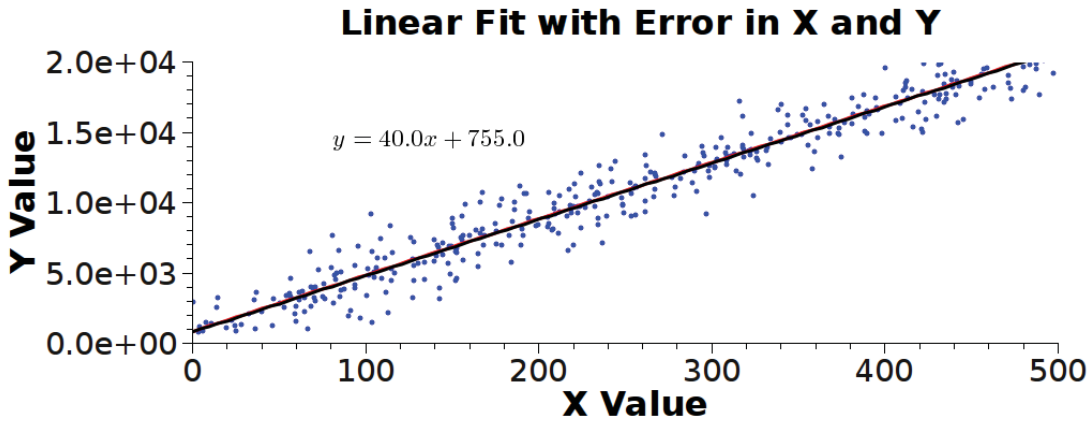
Results and Discussion

Ignoring the uncertainty in the determination of the x coordinate can introduce severe errors in a linear fitting process. As an example, Figure 5.1 shows the effects of using different linear fitting strategies on a synthetic dataset in which an uncertainty in the estimation of the x coordinate has been introduced. The dataset in the figure was created by generating 500 random x, y pairs that satisfy the equation $y = 40.0x + 800$. The x and y coordinate of each pair were then randomly perturbed to simulate an uncertainty in their estimation. Two different linear fitting algorithms were applied to the dataset. The plot on the left shows the results given by a standard algorithm that ignores the error in x but honors the one in y. The plot on the right shows instead the results obtained with an algorithm that honors both errors. Even a visual inspection is sufficient to reveal that the second algorithm returns the values of the function parameters with a higher accuracy. This is confirmed by the examination of the numerical output of the two algorithms.

Obviously, the iterative nature of this algorithm makes it slower that the one which ignores the x coordinate error. However, many important IPLT procedures use it , and in the framework of their complex processing pipeline, the performance loss is negligible.



Errors in y coordinate only



Errors in both coordinates

Figure 5.1 Different linear fitting algorithms applied to a synthetic dataset with errors in the x and y coordinates. The plot on the left shows the results of an algorithm that only honors the uncertainty in the estimation of the y coordinate. The plot on the right shows the output of an algorithm that considers uncertainty in the estimation of both coordinates. In both plots, the red line represents the original function used to create the data ($y = 40.0x + 800$), while the result of the fitting algorithm is shown as a black line. The second algorithms obviously recovers the original parameters used to generate the data with a higher accuracy.

2 LatticeDistortionOverlay: Distorted Lattices

Introduction and Background

Ideally, crystalline samples from electron crystallography should feature a perfect 2-dimensional lattice. Several issues, however, can introduce irregularities and deviations form the perfect conditions: defects the crystal structure, cracks and fragmentation of the crystal, imperfect adhesion of the sample on the supporting carbon film, interaction of the specimen with the electron beam, just to name a few. Although sample preparation and imaging conditions can always be refined until the data quality reaches an appropriate level, the use of a computational correction procedure is preferable when defects are not extensive. A detailed description of the issues and of the challenges of lattice defect correction is beyond the scope of this manuscript (Schenk et al., 2009). It is worth pointing out, however, that the task of analyzing the data is made easier by the ability to visualize graphically all the relevant information. The Lattice Distortion Overlay was designed exactly to meet this need.

Implementation

(<site_alg>/ex2/gui/lib/distorted_lattice_overlay.hh)

The Lattice Distortion Overlay must be used with IPLT's graphical interface GIPLT. The overlay is designed to be superimposed on a correlation map coming from a distortion search. All the information about the theoretical lattice and the distortion of the real crystal is passed during initialization. The overlay can then be added to any open IPLT viewer and it can visualize the theoretical lattice and the structure of the real crystal. It can show the positions of all the unit

cells that were detected in the experimental data and the best estimation for the location of undetected ones. Hovering the mouse pointer around the position of a unit cell allows the user to see its lattice indexes. The algorithm can also show the deviation of the real crystal from the theoretical lattice in the form of distortion vectors connecting the theoretical position of the unit cells with their real location. The representation of the distortion vectors can be scaled to allow an easy visualization of minute distortions.

The overlay was coded using the wxWidgets graphical interface library. This guarantees a perfect integration with all the other elements of GIPLT, which was written using the same development kit(wxWidgets, 2009).

Results and Discussion

A snapshot of the Lattice Distortion Overlay in action can be seen in Figure 5.2

It's worth pointing out that IPLT includes an image processing module that can automatically detect deviations of a crystalline sample from an ideal theoretical lattice. Provided with the expected position of the lattice unit cells, IPLT can detect their real location in the experimental image. If the difference is not extreme, IPLT can also trace the shape of the crystal distortion and use it to perform additional advanced searches(Schenk et al., 2009). The distorted lattice overlay was designed to work in close cooperation with the algorithm that was just described, as it allows the user to visualize all the relevant data at any stage of the processing.

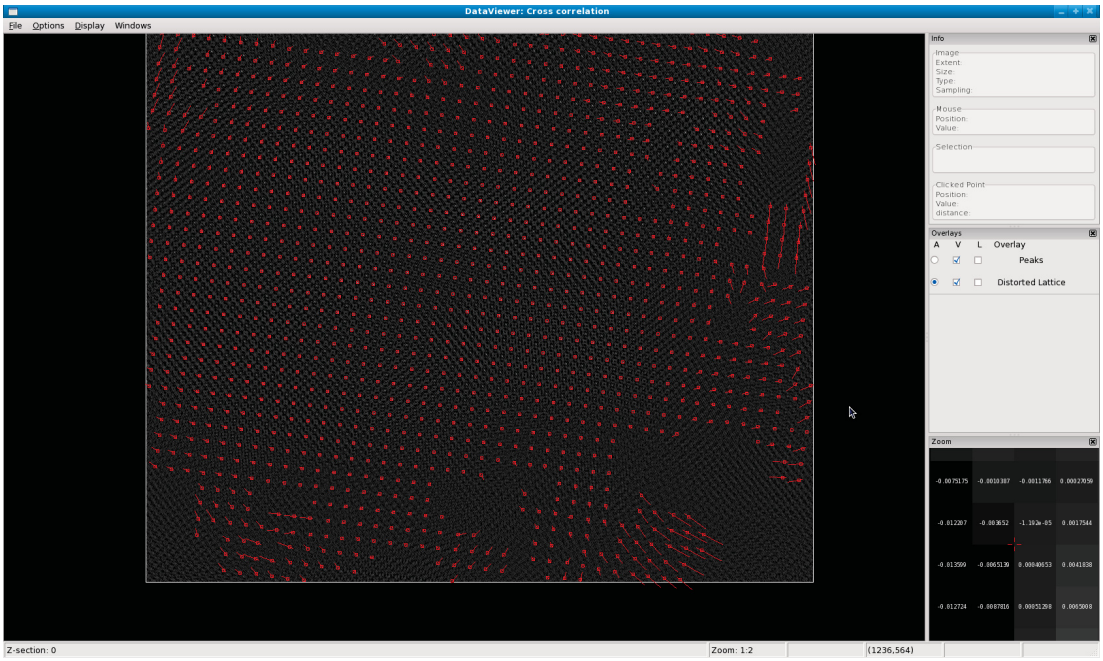


Figure 5.2 The Distorted Lattice Overlay can visualize all unit cells in the theoretical lattice and their position in the experimental crystal. The overlay can also show estimations of the positions of undetected unit cells (This feature is not shown in the image). Distortion vectors (red in the image) can be used to represent the deviation of the real crystal from the theoretical one.

3 CircularMask: Circular Masks in Real Space

Introduction And Background

Using IPLT, image masks can easily be created, stored, manipulated and applied in both real and Fourier space. Their shape can be defined using a series of points called nodes, whose number and positions can be chosen by the user. This approach allows the generation of masks with complex and elaborate shapes. However, sometimes a simple circular mask in real space is needed. The CircularMask algorithm was developed to satisfy this particular need.

Implementation

(<iplt>/src/alg/common/lib/circular_mask.hh)

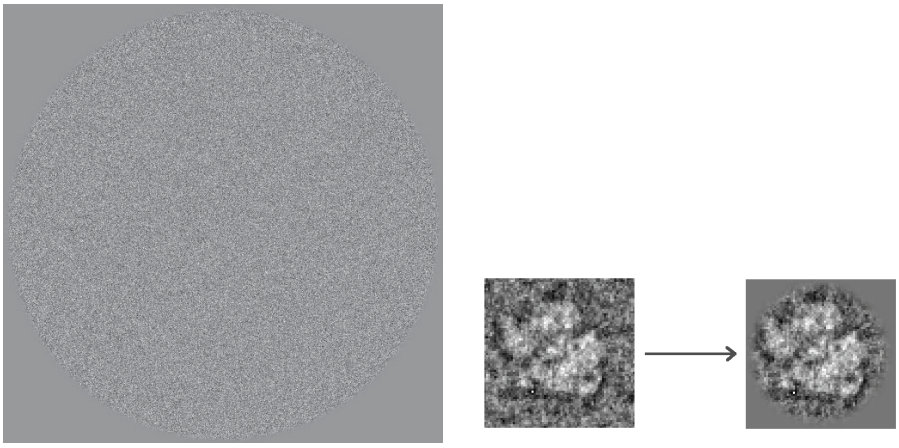
This algorithm applies a circular mask to an image, keeping the content that lies within the circle and discarding what falls outside. The edge of the mask is smooth and has an exponential falloff. The center of the mask and the radius of the circle can be defined when the algorithm is initialized. The same is true of the σ value that defines the size of the falloff. The user can also choose a value that is used to fill the pixels that are masked out.

When one of more of these parameters are not provided, the algorithm chooses some typical default settings. In particular, the largest circle that that can be fitted inside the image is used to define the mask (in other words, the mask is centered in the middle of the image, and the circle has a diameter corresponding to the shortest edge of the image, minus the falloff at both sides) and a value of 3 pixels is chosen for sigma. When no fill value is provided, The algorithm uses the average value of the pixels of the original image.

Internally, the algorithm is very simple but extremely efficient. Away from the edge, a binary mask is applied and the content of the pixels is either accepted or rejected. For pixels that lie close to the edge of the mask, a value is computed by interpolating along a radial Gaussian falloff slope between the original value of the pixel and the masked-out fill value.

Results and Discussion

The CircularMask algorithm is a simple but extremely useful image processing tool. During the manipulation of images, several situations can arise where the application of a straightforward circular mask is required. For example, in some cases images need to be rotated. When images are square or rectangular, precautions must be taken in dealing with the areas near the corners. However, when the information lying in those areas is not important, a circular mask can be applied, turning the actual data content of the image into a round shape, more suited to rotational transformations. In other cases, the relevant information content of an image is restricted to a region surrounding the center (for example, in the case of particle images extracted from a micrograph). Under these circumstances it is preferable to remove as much of the unnecessary background as possible. This can easily be done using a circular mask centered on the area of interest.



Default circular mask Particle masking

Figure 5.3 Possible applications of the CircularMask algorithm. On the left, the circular mask is applied to a simulated carbon film image using default settings. On the right, the algorithm is applied to a small particle image to mask out the background. (Image copyright by Joachim Frank)

4 SmoothMaskImage: Smooth-edged Masks

Introduction and Background

As already stated above, IPLT can easily create, manage and apply image masks, whose shape is defined using a series of reference points called nodes. Normal IPLT masks, however, have sharp edges. While this can be sufficient for many processing tasks, sharp edges in real space images tend to introduce unwanted distortions in Fourier space (Goodman, 2005). This algorithm allows the user to apply a mask that has a smoothed edge with a Gaussian falloff.

Implementation

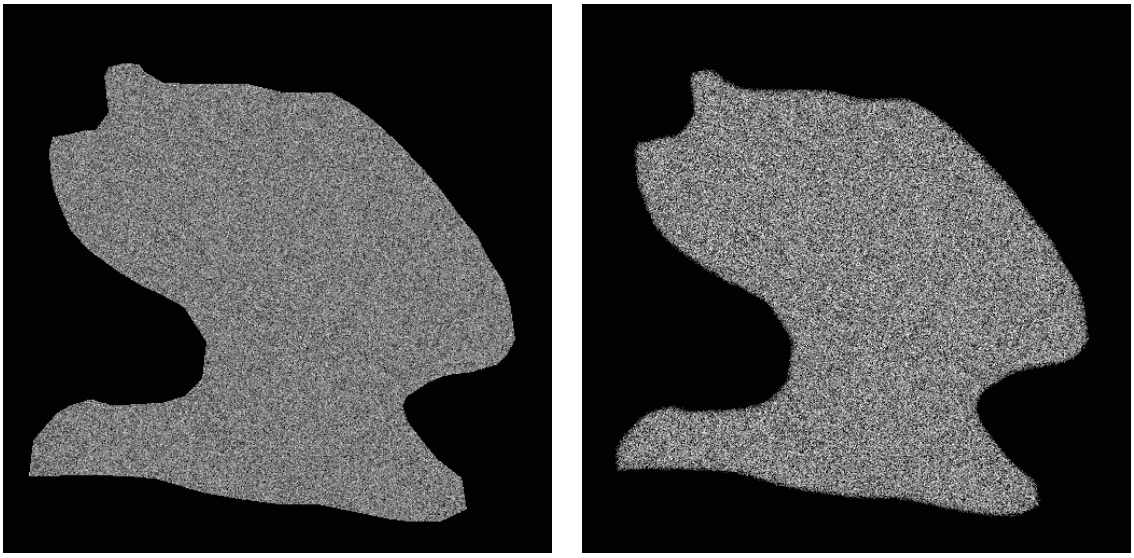
(<iplt>/src/alg/common/lib/smooth_mask_image.hh)

This algorithm does not create a mask, but requires the user to provide one created with other IPLT algorithms. A value of σ that defines the size of the Gaussian falloff must also be chosen. The algorithm can then be applied to an image. This operation is equivalent to applying the original mask on the image, but with a smooth falloff at the borders. The algorithm assigns to the masked out pixels the average value of the pixels of the original image, unless the user opts to fill those pixels with a value of his choice. Internally, the algorithm works by generating a binary image using the input mask, applying a Gaussian filter to it, and multiplying pixel-by pixel the filtered mask image with the image on which the algorithm must be applied.

Results and Discussion

Since this algorithm relies on IPLT for the creation and the management of image masks, all the advantages and features of IPLT masking algorithms are still fully

available to the user. The algorithms just adds the possibility to apply masks with smooth edges. It's worth pointing out that the algorithm uses a real-space Gaussian filter to smooth internally the mask image (Young and Vliet, 1995). This results in a considerable speed gain compared with other approaches which require multiple Fourier transforms.



Normal IPLT mask Mask with smooth edges

Figure 5.4 Effects of a normal IPLT mask (left) and of a mask with smooth edges applied using the SmoothImageMask algorithm (right). For a direct comparison, both masks have been applied to the same simulated carbon film image.

5 ExtractReflection: Extraction of Reflection Data

Introduction and Background

When the microscope in used in imaging mode on a crystalline sample, the reflection peaks in the Fourier transform of the collected images contain both amplitude and phase information. In order to process the data, this information needs to be extracted, and specific amplitude and phase values need to be assigned to each reflection. In ideal conditions (perfect periodicity and infinite extent of the sample), the reflection peaks would be infinitely sharp, resembling δ functions in Fourier space with an associated complex value. In real experiments, however, these conditions are not met, and the information carried by the reflection is spread in an area of Fourier space surrounding the theoretical spatial frequency. Furthermore, image processing is carried out on discretized data, and the spatial frequency of the reflection often falls between Fourier space pixels. This algorithm has been designed to recover the amplitude and phase value of an experimental reflection, allowing at the same time an estimation of the level of background noise inherent in the data.

Implementation

(<site_alg>/ex2/gui/lib/distorted_lattice_overlay.hh)

This algorithm was designed to replicate the process implemented by the TTBOX program in the MRC package (Crowter et al., 1996). A small image containing the are of Fourier space surrounding the reflection (peak box) is passed to the algorithm. The reflection information is assumed to be spread mainly in the four pixels that surround the exact theoretical spatial frequency of the reflection

in 2-dimensions. The complex values of the four pixels are read out, weighed and summed. The weighting coefficient used for each pixel is defined by a sinc function with the following form:

$$Sinc(x) = \frac{Sin\,(\pi(x-q))}{\pi(x-q)}$$

Where x is the frequency coordinate of the pixel for which the coefficient is computed, and q is the exact theoretical frequency of the coordinate. The amplitude and phase of the complex sum is assigned to the reflection.

An accurate estimation of the amplitude value associated with the reflection requires the average background noise level to be subtracted. The computes the average amplitude of the background by summing the values of the pixels lying on the edges of the peak box. Amplitude, phase and background value of the reflection are finally returned to the user. While TTBOX computes interpolated sub-pixel values to obtain an edge that is symmetrical around the theoretical frequency of the reflection, this reimplementation prefers to avoid interpolation and read directly the values of the peak box pixels.

Results and Discussion

The algorithm follows the most recent implementation of the peak extraction routine available in MRC. An examination of the source code of the TTBOX program reveals that different strategies were used in the past to extract the same information and the code that implements them is still present. One of those alternative is also offered to the users of the new reimplemented algorithm, under the name of "OldStyle" variant. In this variant the amplitude of the reflection is not computer from the weighted sum of the complex values of the pixels. Instead, it is computed by summing the amplitudes of the pixels in a 3x3 grid surrounding the theoretical frequency of the reflection.

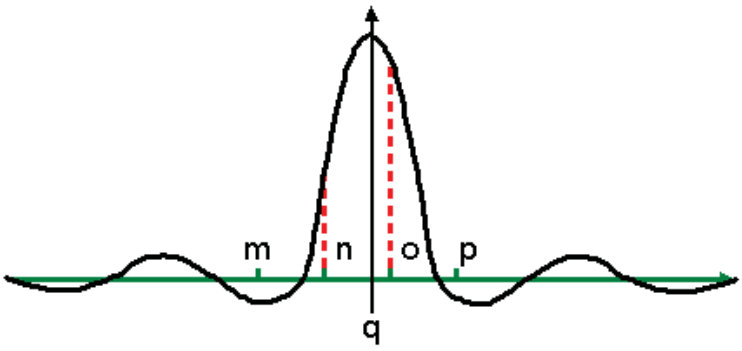


Figure 5.5 Grahical representation of the scaling of pixel data in the ExtractReflection algorithm. The figure shows a 1-dimensional case, but the concept can be easily extended to 2 dimensions. q represents the exact spatial frequency of the reflection. m, n, o and p represent frequencies corresponding to the centers of Fourier space pixels. The relative weight used for each pixel is determined by considering a Sinc function centered on q, and taking as coefficient the function value corresponding to the frequency of the center of the pixel.

6 Envelope: Effects of Instrument's Incoherence

Introduction and Background

Both the classical CTF and the TCIF models of the imaging process assume that the instrument's emissions are perfectly coherent. In real-life expriments, this is obviously not a realistic condition as the electron source is not punctual, but defined by a discrete illumination aperture. Also, the electron beam is not monochromatic and electrons with energies spread over a range of wavelengths are emitted. For the CTF, this phenomena cause the introduction of an additional envelope term in imaging function expression. The effect of this term is the quenching of the signal at high spatial frequencies:

$$Q(\boldsymbol{p}) = -2\sin W0(\boldsymbol{p})E(\boldsymbol{p})$$

The envelope term has a spatial component, given by the finite size of the illumination aperture, and a temporal component, given by the energy spread of the emitted electrons:

$$E(\boldsymbol{p}) = Es(\boldsymbol{p})ET(\boldsymbol{p})$$

The spatial term has the form:

$$Es(\boldsymbol{p}) = e^{-\left(\frac{\pi\lambda p^2 C_c \frac{\Delta E}{E0}}{4\sqrt{\ln 2}}\right)^2}$$

Where ΔE is the energy spread of the incoherent electron beam, $E_0$is the theoretical acceleration voltage and C_c is the chromatic aberration constant of the instrument.

The temporal component has instead the form:

$$Es(\boldsymbol{p}) = e^{-\frac{\left(\pi C s \lambda^2 p^3 - \pi z_0 |p|\right)^2 \alpha_i^2}{\ln 2}}$$

Where α_i is the illumination aperture in radians.

For images collected in tilted geometry (TCIF model), the temporal envelope term has the same form and can be added as an additional teem to the TCIF model, but this is not true for the spatial envelope term. The expression shown above assumes constant defocus across the width of the image. The dependence of the spatial envelope term on the local value of defocus at each point of the sample function makes the derivation of an expression for tilted specimens extremely difficult. An analytical expression for the spatial envelope term is hence not available. This algorithm has been designed to apply the effects of the temporal envelope term to simulations of tilted images.

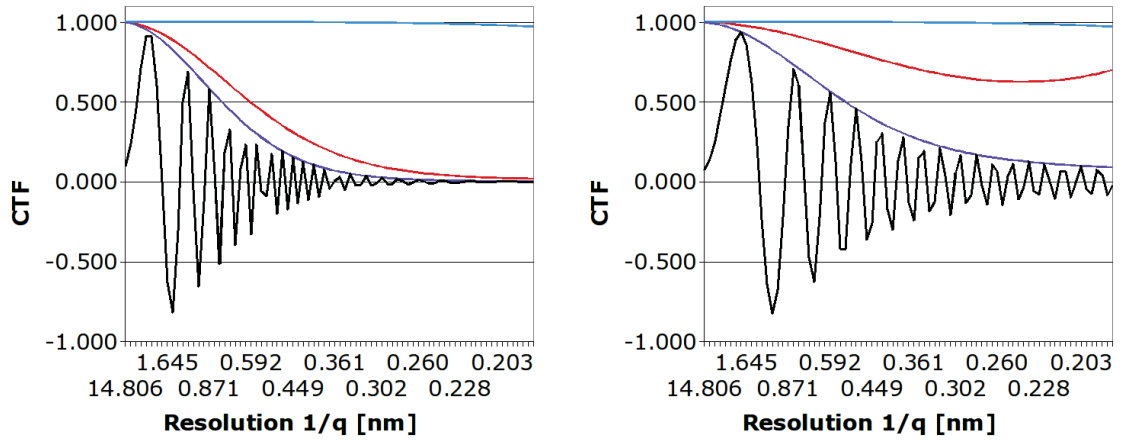
Implementation

(<site_alg>/valerio/envelope/lib/envelope.hh)

This algorithm applies to an image the effects of the temporal envelope term described in the previous section. The information about the incoherence of the instrument's emissions is provided to the algorithm at initialization, together with all the other parameters needed to compute and apply the envelope term (Defocus at the center of the image, wavelength, Spherical aberration constant, etc.). The algorithm can then be applied to any real-space or Fourier space image. Internally, the algorithm will transfer the image into the frequency domain, compute the value of the term at the frequency corresponding to each pixel, and multiply the content of the pixel with the computed value. The algorithms returns then the image to the correct domain, performing a Fourier transform if needed.

Results and Discussions

It should be pointed out the CTF simulation and correction algorithms in IPLT already incorporate the temporal and spatial envelope terms, and take their effects into account. This algorithm is designed to be used on tilted images only and simply applies the effects of the temporal incoherence term. This represents a good approximation of the full incoherence envelope only for imaging conditions under which the temporal envelope is dominant. This generally never happens for typical electron microscopy imaging conditions. Several programs and web applets are anyway available to plot the values of the two terms for a specific set of imaging parameters (For example (CTF Simulation, 2009)). Although these simulations assume untilted geometry and a constant defocus, they can be run several times using imaging conditions that cover the full range of defocus spanned by the tilted image. This can give the user a good idea of the relative magnitude of the two terms and allow the user to judge the reliability of a simulation that only includes the temporal one.



Further from focus

Closer to focus

Figure 5.6 Plots showing the relative weight of the two envelope terms at defocus distances corresponding to two opposite edges of a tilted image, collected under typical electron microscopy imaging conditions. The temporal envelope term is shown in blue on the plot, while the spatial one appears in red. The total envelope, given by the combination of the two, is represented as a violet line. It can be seen that the effect of the spatial envelope is dominant on the effect of the temporal envelope across the whole range of defocus spanned by the image. The plots were computed for a 60 degrees tilted image of size 4000x4000 pixels, with a pixel width of 2 Angstroms. The defocus distance in the center of the image was set at 2000 nm, bringing the edge closer to the image at a defocus distance of 2346 nm, an the edge lying further from focus at a defocus distance of 2346 nm. The illumination aperture angle was set at 0.1 mrad and the energy spread of the emitted electrons at 0.03eV. The values of the other parameters used in the simulation were: acceleration voltage 200kV, spherical and chromatic aberration constants 2mm

7 Annex: Multi-Resolution Modeling in IPLT

The IPLT software framework was developed to process electron crystallography data, with the final goal of being able to recover the 3-dimensional structure of a sample. But IPLT was designed mainly to manipulate 2-dimensional images: it deals with all the important processing procedures, but leaves the task of computing and visualizing the final structure to other software packages. It should be pointed out, however, that nothing in the software capabilities or in its design philosophy prevents it from being used as a 3-dimensional processing tool. As a proof-of-concept, IPLT was used to develop a internal docking application. While the application itself just features basic functionality, has not been fully tested, and has never been used in a real scientific project, it serves as a demonstration of the flexibility and expandability of IPLT, and testifies to the strength of the design philosophy that lies at its base. This section of the manuscript will briefly introduce the nature of the internal docking process, shortly discuss its implementation in IPLT, and present all the algorithms that were developed to perform the required tasks.

Background

Although all fields of electron microscopy underwent strong development in recent years, the level of resolution of x-ray techniques still remains out of reach, even if just barely (Zhou, 2008). Electron microscopy techniques have nevertheless some strong advantages: they can be used to image structures bigger than a simple protein, like molecular machines and nanostructures (Frank, 1996), and they allow the examination of specimens in an environment that is much closer to the biological native one (Frank, 2006). Ideally, one would like to be combine these advantages with the high resolution granted by x-ray crystallography. Although it is not possible to do it at the experimental level, interesting results can be achieved

by combining the two approaches computationally. Sometimes crystallographic structures of proteins that belong to big molecular machines are available, and electron microscopy provides a density map at lower resolution for the whole assembly. If the location of the single protein in the big agglomerate can be detected, high resolution data can be plugged directly into the density map. When this process is repeated for each protein in the assembly, a detailed model of the molecular machine is built *in silico*. This technique takes the name of multi-resolution modeling, and the procedure to determine the position of the protein is called internal docking.

It should be pointed out that the conditions under which the structure and the density map have been obtained are radically different. In particular, since the protein structure is collected using x-ray techniques, some crystal packing artifacts could be present, and the structure itself could be considerably distorted from its native shape. This often prevents a perfect match with the low resolution density map, and renders the internal docking procedure impossible without applying some sort of deformation. When the magnitude of this problem is limited, however, the small protein can be considered as a rigid body, and the internal docking procedure becomes conceptually simple. A scoring function that measures the matching of the high resolution structure with the lower resolution density is defined. Then all the possible ways in which the protein structure can be placed in 3-dimensional space are to be explored, until the one that maximizes the matching is found.

The high resolution structure, which is called probe structure in this manuscript, usually comes in the form of a list of atom types with their location in space (Berman et al., 2003). This structure is projected on a density map, and the resolution is lowered at the same level of the electron microscopy density map (which is called in this manuscript target structure). The coefficient of cross-correlation between the two density maps is then used as a scoring func-

tion (Volkmann and Hanein, 1998). All possible rotations and translations of the probe structure are explored, and the conditions giving the highest correlation value are taken as the correct placement. In order to speed up the exploration of translational space, the Katchalski-Katzir correlation theorem and FFT techniques are applied whenever possible (Katchalski-Katzir et al., 1992). Rotational and translational space cannot obviously be explored with arbitrary accuracy: sampling steps have to be chosen, and this limits the precision with which the correct placement is determined. The best solutions can however be refined later, performing a new search with finer sampling steps. A detailed analysis of the challenges posed by multi-resolution modeling projects and an assessments of the intrinsic limits of correlation-based internal docking are beyond the scope of this manuscript. Good reports on the advancement of this field have been published in recent years and the reader is referred to them (Wriggers and Chacon, 2001 and Heyd and Birmanns, 2008).

Internal Docking and IPLT

The full internal docking processing pipeline has been implemented in IPLT. IPLT relies on the external package CCP4 (Collaborative Computational Project, Number 4, 1994) to rotate and translate atomic structures, generating bash scripts that call external standalone programs. CCP4 programs are also used to project the atomic structure on a density map. All the remaining steps, the exploration of rotational and translational space, the detection and management of correlation coefficient maxima, and the optimization of the best solutions, are performed internally. Some other preparation steps, like the generation of a an efficient sampling of rotational space, are also performed within IPLT. Only a limited number of new algorithms needed to be developed to implement the whole procedure.

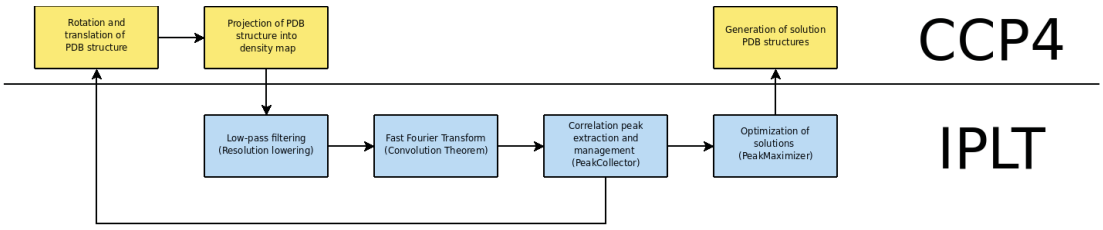


Figure 5.7 Diagram of the internal docking procedure in IPLT. The figure shows which parts of the process were performed using internal IPLT algorithms, and for which parts CCP4 external programs were needed.

8 GenerateRotationCoverage: Rotational Space

Introduction and background

In correlation-coefficient based internal docking the whole translational and rotational spaces need to be explored. While a regular and efficient sampling of translational space can be achieved using a simple Cartesian reference system, obtaining the same result for rotational space is not trivial. As a first step, a notation to describe orientations in rotational space must be chosen. Euler angles have been used in recent years in structural biology to achieve this goal. Euler angles come in triplets, φ, θ, and ψ. Each angle describes a rotation around an axis. Initially, a rotation with angle φ around the z-axis of the reference system is applied. Then follows a rotation with angleθ around the x-axis. This separates the z-axis of a system embedded in the rotated object from the z-axis of the original absolute reference system. The third rotation, with an angle ψ, is then performed around the z-axis embedded in the object (See Figure 5.8). This is the so-called $z - x - z$ convention. Other conventions exists, in which the rotational axes or the order of the rotations vary, but all follow the same general principle (Arfken, 2005). Using Euler angles, any arbitrary rotational orientation can be described with just 3 parameters.

Rotational space can be explored by generating a list of Euler angles that samples it with sufficient accuracy, and then by applying each rotation in the list to the probe structure placed in a standard initial orientation. Having the Euler angles φ and ψ vary in the range from 0 to 2π radians and the angle θ varying in the range from 0 to π covers the whole rotational space, however some care must be taken in choosing the values that the angles can assume within their validity ranges: obtaining a regular sampling is not straightforward.

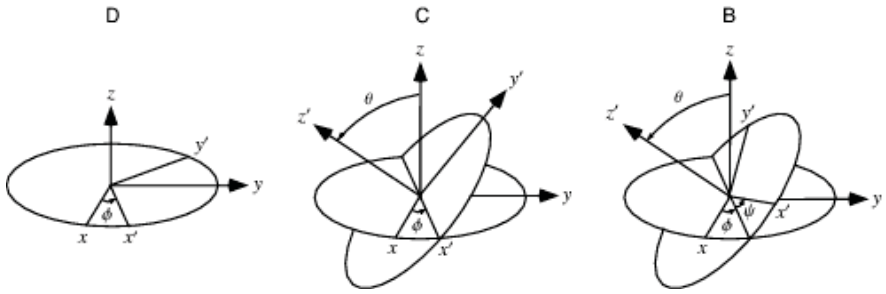


Figure 5.8 The effects of an arbitrary rotation in 3-dimensional space can be described using a triplet of Euler angles. This figure shows a schematic representation of the Euler angles according to the z-x-z convention. They describe three rotations that, applied in succession, transform the absolute reference system into the rotated one. (Image copyright by Wolfram Research)

Implementation

(<iplt>/src/geom/lib/composite3_op.hh)

A discrete sampling of rotational space can be obtained by having the three angles step through their validity ranges in step of n radians, where n is an arbitrary number which defines the coarseness of the sampling. This, however, introduces an irregularity. When the rotational space is represented as a sphere, the areas surrounding the "poles" (determined by the z axis) have a much finer sampling than the areas around the equator (See Figure 5.9 on the left). This algorithm implements a simple but effective strategy to reduce the population of the polar area. Instead of changing the three angles in regular steps, a number of "latitude circles" are defined on the sphere. The distance between the latitude circles is determined by the angular step n . Each "latitude circle" is then be subdivided in a number "longitudinal segments" of equal angular size. The angular length of these segments is chosen to be as close as possible to the sampling angular step. The points defined on the sphere using this strategy are are finally translated into triplets of Euler Angles. The algorithm can be used to sample a subset of rotational space. This can be achieved by providing limits to the validity ranges of the three Euler angles when the algorithm is initialized. The algorithm returns a list of IPLT Euler angles mathematical objects, but internal IPLT functions can be used to transform them in rotational matrices or quaternions .

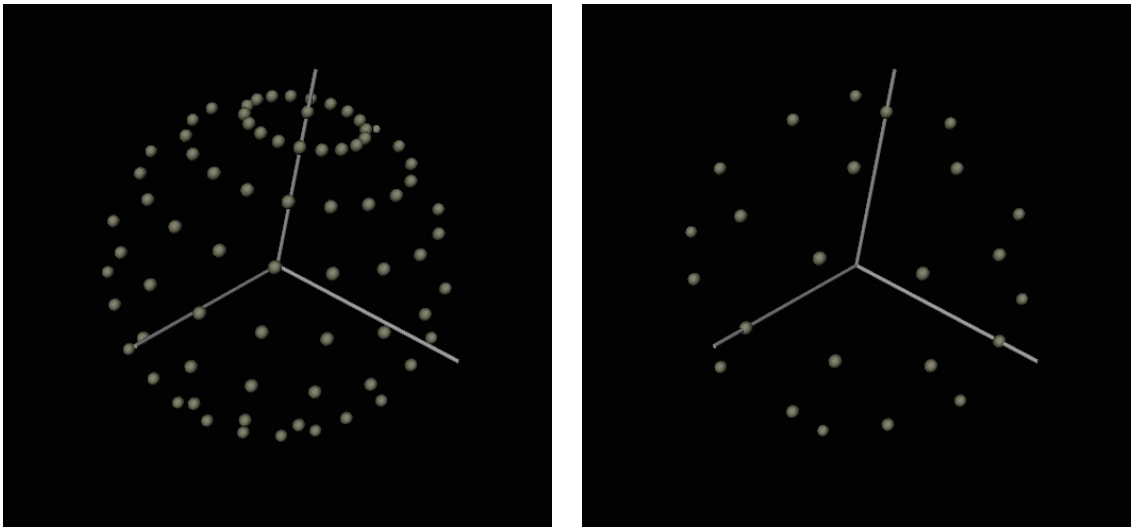
Results and Discussion

This algorithm generates a sampling of rotational space that is reasonably regular. Obviously, a perfect match between the length "latitudinal segments" and the angular sampling step is almost never achieved, as the length of the circumference of the "latitude circle" is in general not a multiple of the angular step. However,

this approach generates a rotational sampling that is sufficiently regular to be used in an internal docking procedure and certainly less irregular that constant angle steps (See figure 5.9).

It should also be mentioned that this strategy avoids the presence of multiple equivalent triplets in the Eulers angle list. This problem arises because several sets of Euler angles can correspond to the same spatial orientation. For example, when the second rotation is null, first and the third rotations happen around parallel axes (singularity). The first and third rotations can then cancel each other, and several triplets can become equivalent. This algorithm, however, avoids the problem by generating the needed orientations first and only translating them into Euler angles at a later stage, thus avoiding some of their drawbacks, like singularities.

It should be noted that several techniques to generate discrete samplings of rotational space using quaternions have been recently proposed. (For example Karney (2007)). These approaches present several advantages over the use of Euler angles: better mathematical tractability, stronger uniformity of the sampling, and lack of singularities). They should be selected for future implementations of this algorithm.



Constant angular steps IPLT algorithm

Figure 5.9 Sampling of rotation space using constant angular steps for the three euler angles and using the IPLT algorithm. Rotational space is represented as a sphere and triplet of Euler angles are shown as dots on the sphere. The method using constant angular steps leads to oversampling of the area around the poles and to sparse sampling of the area surrounding the equator. The IPLT algorithms generates instead a sampling with a higher regularity. Angular step is 20 degrees for both methods. For the sake of clarity, dots lying on the emisphere that faces away from the reader do not appear in the plot.

9 PeakCollector: Correlation Peak Management

Introduction and Background

As stated above, in correlation-based internal docking algorithms it is necessary to explore all possible orientations and translations of the probe structure and obtain a correlation coefficient value for each case. While all possible orientations must be enumerated and evaluated one by one, the Katchalski-Katzir correlation theorem (Katchalski-Katzir et al., 1992)can be used in conjunctions with FFT techniques to quickly sample translational space:

$$Cc(\boldsymbol{p}) = \mathcal{F}{-}\infty \left\{ \mathcal{F}\left[\rho(\boldsymbol{p})\right] \cdot \mathcal{F}\left[\mathcal{R}(\boldsymbol{p})\right] \right\}$$

In this expression, $\mathcal{F}$ represents a Fourier Transform, $\cdot$ represents a multiplication operator. $\rho(\boldsymbol{p})$ represents the density function of the probe structure, $R(\boldsymbol{p})$ is the density function of the target structure, and C_c represents the cross-correlation coefficient. The result of the application of the theorem for a specific orientation is a 3-dimensional correlation map that covers all possible translations. Translations for which the correlation coefficient assumes particularly high values appear as peaks in the map are candidates for further refinement. Algorithms are available in IPLT to extract peaks from correlation maps, however some further processing needs to applied. Refinement of docking solutions is computationally expensive, and is usually performed only for the most promising candidates. Hence, it would be optimal for the list of extracted peaks to be sorted in order of decreasing correlation coefficient values. Furthermore, clusters of correlation peaks which feature very similar roto-translational parameters need to be analyzed and resolved. This algorithm was designed to perform all these tasks.

Implementation

(<site_alg>/valerio/docking/lib/peak_collector.hh)

Provided with a series of correlation peaks, this algorithm returns a list of peaks sorted in order of descending value of the correlation coefficient. When the algorithm is initialized, the maximum number of peaks in the list and the minimum translational distance for two peaks to be considered separate (exclusion radius) must be provided provided. Peaks are then fed to the algorithm, in the form of translation coordinates with an attached correlation value. Internally, the algorithm processes the input according to the following checklist:

- if the newly entered peak is outside of the exclusion radius of all other peaks, it is added to the list
- if the newly entered peak falls within the exclusion radius of another peak in the list and its value is higher than the peak generating the exclusion radius, it is added to the list, and the peak generating the radius is removed.
- if the newly entered peak falls within the exclusion radius of another peak in the list but its value is not higher than the peak generating the exclusion radius, it is ignored.
- If a peaks get added to the list and the maximum number of peaks in the list is exceeded, the peak with the lowest correlation value is removed.

Results and Discussion

This algorithm performs all the bookkeeping necessary to generate a sorted list of the strongest peaks in the correlation map. It should be noted, however, that the algorithm does not detect the peaks or extracts them from the map. This task is

left to the user, who must provide the algorithm with its input data. The algorithm employs a clever and efficient processing strategy that avoids unnecessary sorting and ordering of the internal data, resulting in a high operational speed. Furthermore, it processes peak data as soon as they are provided, and does not require a large amount of memory to store internal information. The algorithm can automatically resolved clusters of sub-peaks that sometimes appear around a strong correlation peak. All the sub-peaks have very similar roto-translational coordinates and represents small perturbations of the same conditions. Only the strongest representative peak from each cluster is selected by the algorithm. The algorithm allows the user to chose the maximum translational distance the two peaks can have in order to be considered members of the same cluster.

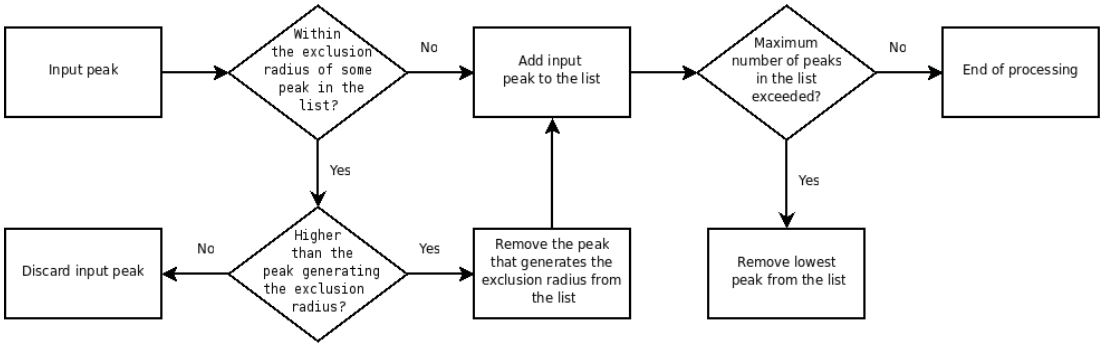


Figure 5.10 Flowchart diagram of the logic implemented by the PeakCollector algorithm.

10 PeakMaximizer: Correlation Peak Optimization

Introduction and Background

Internal docking procedures require an extensive exploration of rotational and translational spaces. Obviously, the exploration cannot be infinitely precise: it takes place in discrete translational and rotational sampling steps, and their size determines the accuracy of the search. Once the best set of coordinates has been identified, it is possible to increase the accuracy with which it was determined by exploring the surrounding roto-translational space with finer sampling steps. This advanced exploration does not happen through an extensive enumeration of all possible combinations. Instead, it assumes that a solution with a higher coefficient, if it exists, lies close to the current best estimation and the the likelihood of a local maximum being present between the two is low. This the best solution is then reached using a maximization algorithm (Chacón and Wriggers, 2002). This algorithm was developed to refine an internal docking solution using a maximization approach.

Implementation

(<site_alg>/valerio/docking/lib/peak_maximizer.hh)

This algorithm refines a rough solution of an internal docking problem, determining with higher accuracy the set of coordinates that maximizes the scoring function. The density maps of the probe and of the target structures (See section X) must be provided at initialization, together with the rotational and translational coordinates of the solution to be optimized. The base sampling step of the optimization must also be chosen when the algorithm is initialized. After

the processing is complete, the algorithm returns the translational and rotational coordinates of the optimized solution.

Internally the algorithm employs a strategy based on the downhill simplex method (Nelder-Meads method) (Nelder and Mead, 1965), and minimizes the negative of the scoring function with respect to the roto-translational parameters. The Nelder-Meads method searches an n-dimensional space using the shape of a polytope with n+1 vertexes (simplex) as a guide . At each step, the simplex vertex where the value of the function is highest is replaced with the its reflection through the centroid of the polytope, according to one of the many available reflection strategies (See Figure 5.11). The process is iteratively repeated until no change of vertex results in a lower value of the scoring function, as a minimum (local or global) has been reached. The peak maximization algorithm uses the implementation of the Simplex method available in the Gnu Scientific Library (Gnu Scientific Library, 2009). Helper functions have been added to manage the input output of data and to integrate the GNU routines into a C++ object-oriented programming environment.

Results and Discussion

The quality of a solution of an internal docking problem is difficult to assess. It is even known that sometimes the solution featuring the highest correlation coeffcient is not the one giving the best structural overlap (Volkmann and Hanein, 1998). Visual inspection, and compatibility with known biological features of the sample are often the only criteria that can be used to judge the quality of a solution. In this respect, the optimization of a rough solution can improve its quality in a dramatic way. It should however be pointed out that the algorithm has some important limitations. As all other minimization strategies, this algorithm can get stuck in a local maximum. The assumption that the optimization route s

free of local maxima is not verified and could turn out not to be true for specific cases. Furthermore, an anomalous step can trigger the simplex algorithm's stopping conditions, even when no maximum has been reached. It is then customary to restart the algorithm to verify that no further optimization is really possible (Vetterling and Flannery, 2002). As with other computational methods, the user should never blindly trust the results of the algorithm. Care must be taken into verifying that the solution that was proposed by the optimization routine is really relevant and human judgment often plays a role in this process.

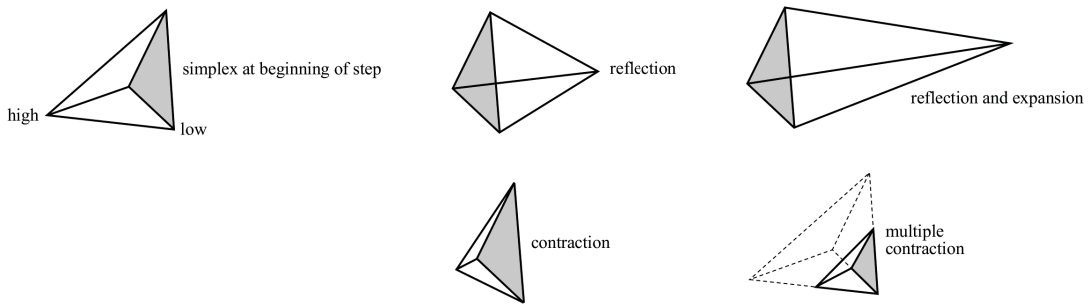


Figure 5.11 Reflection strategies employed by the downhill simplex algorithm. (Image copyright Cambridge Univeristy Press)

Appendix 1 : Derivatives for the Fitting of TCIF Parameters

This appendix contains the full expressions for the derivatives of equation 2.25 in Chapter 2. This part of the manuscript has been created using the Mathematica program (Mathematica, 2009) : the numbering of the equations will be internally consistent but indipendent from the rest of the manuscript. The graphical layout will also feature some slight differences.

The mathemetical notation also shows some differences. The conversion is straightforward when the following guidelines are kept in mind:

$$p = (\mathrm{px}, \mathrm{py}) \tag{1}$$

$$\mu_n = (\mathrm{mx}_n, \mathrm{my}_n) \tag{2}$$

Full expression for equation 2.25 in Chapter 2:

$$\begin{aligned}
|Q(p)|^2 = {} & e^{-\frac{1}{2}\pi^2(\mathrm{px}^2+\mathrm{py}^2)\sigma^2(4-(\mathrm{px}^2+\mathrm{py}^2)\lambda^2+(\mathrm{px}^2+\mathrm{py}^2)\lambda^2\,\mathrm{Sec}[\alpha]^2+4\lambda(\mathrm{px}\,\mathrm{Cos}[\beta\,\mathrm{XXX}]+\mathrm{py}\,\mathrm{Sin}[\beta])\,\mathrm{Tan}[\alpha])} \\
& \left(\left(\sum_{n=1}^{M}\mathrm{Cos}\left[2\pi\left(\mathrm{mx}_n\left(\mathrm{px}+\frac{1}{2}(\mathrm{px}^2+\mathrm{py}^2)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\mathrm{my}_n\left(\mathrm{py}+\frac{1}{2}(\mathrm{px}^2+\mathrm{py}^2)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\right]\right.\right. \\
& \left.\left.\mathrm{A}_n\right)^2 + e^{4\pi^2(\mathrm{px}^2+\mathrm{py}^2)\lambda\sigma^2(\mathrm{px}\,\mathrm{Cos}[\beta]+\mathrm{py}\,\mathrm{Sin}[\beta])\,\mathrm{Tan}[\alpha]} \\
& \left(\left(\sum_{n=1}^{M}\mathrm{Cos}\left[2\pi\left(\mathrm{mx}_n\left(\mathrm{px}-\frac{1}{2}(\mathrm{px}^2+\mathrm{py}^2)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\right.\right.\right.\right. \\
& \left.\left.\left.\left.\mathrm{my}_n\left(\mathrm{py}-\frac{1}{2}(\mathrm{px}^2+\mathrm{py}^2)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\right]\mathrm{A}_n\right)^2 + \\
& \left(\sum_{n=1}^{M}-\mathrm{Sin}\left[2\pi\left(\mathrm{mx}_n\left(\mathrm{px}-\frac{1}{2}(\mathrm{px}^2+\mathrm{py}^2)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\mathrm{my}_n\right.\right.\right. \\
& \left.\left.\left.\left(\mathrm{py}-\frac{1}{2}(\mathrm{px}^2+\mathrm{py}^2)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\right]\mathrm{A}_n\right)^2\Bigg) + \\
& \left(\sum_{n=1}^{M}-\mathrm{Sin}\left[2\pi\left(\mathrm{mx}_n\left(\mathrm{px}+\frac{1}{2}(\mathrm{px}^2+\mathrm{py}^2)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\mathrm{my}_n\left(\mathrm{py}+\frac{1}{2}(\mathrm{px}^2+\mathrm{py}^2)\right.\right.\right.\right. \\
& \left.\left.\left.\left.\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\right]\mathrm{A}_n\right)^2 - \\
& 2\,e^{2\pi^2(\mathrm{px}^2+\mathrm{py}^2)\lambda\sigma^2(\mathrm{px}\,\mathrm{Cos}[\beta]+\mathrm{py}\,\mathrm{Sin}[\beta])\,\mathrm{Tan}[\alpha]}\left(\left(\sum_{n=1}^{M}-\mathrm{Sin}\left[2\pi\left(\mathrm{mx}_n\left(\mathrm{px}-\frac{1}{2}(\mathrm{px}^2+\mathrm{py}^2)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\right.\right.\right.\right. \\
& \left.\left.\left.\left.\mathrm{my}_n\left(\mathrm{py}-\frac{1}{2}(\mathrm{px}^2+\mathrm{py}^2)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\right]\mathrm{A}_n\right) \\
& \left(\mathrm{Sin}[2\,\mathrm{W0}[\mathrm{px},\mathrm{py}]]\sum_{n=1}^{M}\mathrm{Cos}\left[2\pi\left(\mathrm{mx}_n\left(\mathrm{px}+\frac{1}{2}(\mathrm{px}^2+\mathrm{py}^2)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\right.\right.\right.
\end{aligned} \tag{3}$$

$$my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,A_n+$$

$$\text{Cos}[2\,W0[px,\,py]]\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,A_n\Bigg)+$$

$$\left(\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+my_n\right.\right.$$

$$\left.\left.\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,A_n\right)$$

$$\Bigg(\text{Cos}[2\,W0[px,\,py]]\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,A_n-$$

$$\text{Sin}[2\,W0[px,\,py]]\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,A_n\Bigg)\Bigg)\Bigg)$$

Derivative with respect to α:

$$\frac{\partial\,|Q(p)|^2}{\partial\,\alpha}=\frac{1}{2}\,e^{-\frac{1}{2}\,\pi^2\left(px^2+py^2\right)\sigma^2\left(4-\left(px^2+py^2\right)\lambda^2+\left(px^2+py^2\right)\lambda^2\,\text{Sec}[\alpha]^2+4\,\lambda\,(px\,\text{Cos}[\beta]+py\,\text{Sin}[\beta])\,\text{Tan}[\alpha]\right)}$$

$$\Bigg(4\Bigg(2\,e^{4\,\pi^2\left(px^2+py^2\right)\lambda\,\sigma^2\,(px\,\text{Cos}[\beta]+py\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}\,\pi^2\left(px^2+py^2\right)\lambda\,\sigma^2\,\text{Sec}[\alpha]^2$$

$$(px\,\text{Cos}[\beta]+py\,\text{Sin}[\beta])\Bigg(\Bigg(\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$my_n\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,A_n\Bigg)^2+$$

$$\left(\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+my_n\right.\right.$$

$$\left.\left.\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,A_n\right)^2\Bigg)-$$

$$2\,e^{2\,\pi^2\left(px^2+py^2\right)\lambda\,\sigma^2\,(px\,\text{Cos}[\beta]+py\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}\,\pi^2\left(px^2+py^2\right)\lambda\,\sigma^2\,\text{Sec}[\alpha]^2$$

$$(px\,\text{Cos}[\beta]+py\,\text{Sin}[\beta])\Bigg(\Bigg(\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$\left.\left.\left.

$$\sum_{n=1}^{M} -2\,\pi\,\mathrm{Sin}\Big[2\,\pi\Big(\mathrm{mx}_n\Big(\mathrm{px}-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\Big)+ \mathrm{my}_n\Big(\mathrm{py}-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\Big)\Big)\Big]\,\mathrm{A}_n \Big(-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Sec}[\alpha]^2\,\mathrm{mx}_n-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sec}[\alpha]^2\,\mathrm{Sin}[\beta]\,\mathrm{my}_n\Big)\Big)+$$

$$\Big(\sum_{n=1}^{M} -\mathrm{Sin}\Big[2\,\pi\Big(\mathrm{mx}_n\Big(\mathrm{px}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\Big)+\mathrm{my}_n \Big(\mathrm{py}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\Big)\Big)\Big]\,\mathrm{A}_n\Big)$$

$$\sum_{n=1}^{M} -2\,\pi\,\mathrm{Cos}\Big[2\,\pi\Big(\mathrm{mx}_n\Big(\mathrm{px}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\Big)+ \mathrm{my}_n\Big(\mathrm{py}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\Big)\Big)\Big]\,\mathrm{A}_n \Big(\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Sec}[\alpha]^2\,\mathrm{mx}_n+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sec}[\alpha]^2\,\mathrm{Sin}[\beta]\,\mathrm{my}_n\Big)+$$

$$\Big(\sum_{n=1}^{M} \mathrm{Cos}\Big[2\,\pi\Big(\mathrm{mx}_n\Big(\mathrm{px}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\Big)+\mathrm{my}_n \Big(\mathrm{py}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\Big)\Big)\Big]\,\mathrm{A}_n\Big)$$

$$\sum_{n=1}^{M} -2\,\pi\,\mathrm{Sin}\Big[2\,\pi\Big(\mathrm{mx}_n\Big(\mathrm{px}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\Big)+ \mathrm{my}_n\Big(\mathrm{py}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\Big)\Big)\Big]\,\mathrm{A}_n \Big(\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Sec}[\alpha]^2\,\mathrm{mx}_n+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sec}[\alpha]^2\,\mathrm{Sin}[\beta]\,\mathrm{my}_n\Big)-$$

$$e^{2\,\pi^2\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\sigma^2\,(\mathrm{px}\,\mathrm{Cos}[\beta]+\mathrm{py}\,\mathrm{Sin}[\beta])\,\mathrm{Tan}[\alpha]}$$

$$\Big(\Big(\mathrm{Sin}[2\,\mathrm{W0}[\mathrm{px},\,\mathrm{py}]]\sum_{n=1}^{M}\mathrm{Cos}\Big[2\,\pi\Big(\mathrm{mx}_n\Big(\mathrm{px}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\Big)+ \mathrm{my}_n\Big(\mathrm{py}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\Big)\Big)\Big]\,\mathrm{A}_n+ \mathrm{Cos}[2\,\mathrm{W0}[\mathrm{px},\,\mathrm{py}]]\sum_{n=1}^{M}-\mathrm{Sin}\Big[2\,\pi\Big(\mathrm{mx}_n\Big(\mathrm{px}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\Big)+ \mathrm{my}_n\Big(\mathrm{py}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\Big)\Big)\Big]\,\mathrm{A}_n\Big)$$

$$\sum_{n=1}^{M} -2\,\pi\,\mathrm{Cos}\Big[2\,\pi\Big(\mathrm{mx}_n\Big(\mathrm{px}-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\Big)+$$

$$\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big]\Big]\,A_n$$

$$\left(-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Sec}[\alpha]^2\,\text{mx}_n-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sec}[\alpha]^2\,\text{Sin}[\beta]\,\text{my}_n\right)+$$

$$\left(\text{Cos}[2\,W0[\text{px},\,\text{py}]]\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$

$$\left.\left.\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n-$$

$$\text{Sin}[2\,W0[\text{px},\,\text{py}]]\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$\left.\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n\Bigg)$$

$$\sum_{n=1}^{M}-2\,\pi\,\text{Sin}\Big[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$\left.\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n$$

$$\left(-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Sec}[\alpha]^2\,\text{mx}_n-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sec}[\alpha]^2\,\text{Sin}[\beta]\,\text{my}_n\right)+$$

$$\left(\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$

$$\left.\left.\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n\right)$$

$$\left(-\text{Sin}[2\,W0[\text{px},\,\text{py}]]\sum_{n=1}^{M}-2\,\pi\,\text{Cos}\Big[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$

$$\left.\left.\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n$$

$$\left(\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Sec}[\alpha]^2\,\text{mx}_n+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sec}[\alpha]^2\,\text{Sin}[\beta]\,\text{my}_n\right)+$$

$$\text{Cos}[2\,W0[\text{px},\,\text{py}]]\sum_{n=1}^{M}-2\,\pi\,\text{Sin}\Big[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$\left.\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n\left(\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\right.$$

$$\left.\text{Cos}[\beta]\,\text{Sec}[\alpha]^2\,\text{mx}_n+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sec}[\alpha]^2\,\text{Sin}[\beta]\,\text{my}_n\right)\Bigg)+$$

$$\left(\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\right.\right.$$

$$\left.\left.\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n\right)$$

$$\left(\text{Cos}[2\,\text{W0}[\text{px},\,\text{py}]]\sum_{n=1}^{M}-2\,\pi\,\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.\right.$$
$$\left.\left.\left.\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n$$
$$\left(\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Sec}[\alpha]^2\,\text{mx}_n+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sec}[\alpha]^2\,\text{Sin}[\beta]\,\text{my}_n\right)+$$
$$\text{Sin}[2\,\text{W0}[\text{px},\,\text{py}]]\sum_{n=1}^{M}-2\,\pi\,\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$
$$\left.\left.\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n\left(\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\right.$$
$$\left.\left.\text{Cos}[\beta]\,\text{Sec}[\alpha]^2\,\text{mx}_n+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sec}[\alpha]^2\,\text{Sin}[\beta]\,\text{my}_n\right)\right)\right)-$$
$$2\,\pi^2\left(\text{px}^2+\text{py}^2\right)\lambda\,\sigma^2\,\text{Sec}[\alpha]^2\left(\left(\sum_{n=1}^{M}\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.\right.\right.$$
$$\left.\left.\left.\left.\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n\right)^2+$$
$$e^{4\,\pi^2\left(\text{px}^2+\text{py}^2\right)\lambda\,\sigma^2\,(\text{px}\,\text{Cos}[\beta]+\text{py}\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}\left(\left(\sum_{n=1}^{M}\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\right.\right.\right.\right.\right.$$
$$\left.\left.\left.\left.\left.\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n\right)^2+$$
$$\left(\sum_{n=1}^{M}-\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\right.\right.\right.$$
$$\left.\left.\left.\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n\right)^2\Bigg)+$$
$$\left(\sum_{n=1}^{M}-\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}+\right.\right.\right.\right.$$
$$\left.\left.\left.\left.\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n\right)^2-$$
$$2\,e^{2\,\pi^2\left(\text{px}^2+\text{py}^2\right)\lambda\,\sigma^2\,(\text{px}\,\text{Cos}[\beta]+\text{py}\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}\left(\left(\sum_{n=1}^{M}-\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\right.\right.\right.\right.\right.$$
$$\left.\left.\left.\left.\left.\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n\right)$$
$$\left(\text{Sin}[2\,\text{W0}[\text{px},\,\text{py}]]\sum_{n=1}^{M}\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.\right.$$

$$my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big]\,A_n+$$

$$\text{Cos}[2\,W0[px,\,py]]\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,A_n\Bigg)+$$

$$\left(\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+my_n\right.\right.$$

$$\left.\left.\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n\right)$$

$$\Bigg(\text{Cos}[2\,W0[px,\,py]]\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,A_n-$$

$$\text{Sin}[2\,W0[px,\,py]]\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,A_n\Bigg)\Bigg)$$

$$\left(2\,px\,\text{Cos}[\beta]+2\,py\,\text{Sin}[\beta]+\left(px^2+py^2\right)\lambda\,\text{Tan}[\alpha]\right)\Bigg)$$

Derivative with respect to β:

$$\frac{\partial\,|Q\,(p)|^2}{\partial\,\beta}=2\,e^{-\frac{1}{2}\,\pi^2\left(px^2+py^2\right)\sigma^2\left(4-\left(px^2+py^2\right)\lambda^2+\left(px^2+py^2\right)\lambda^2\,\text{Sec}[\alpha]^2+4\,\lambda\,(px\,\text{Cos}[\beta]+py\,\text{Sin}[\beta])\,\text{Tan}[\alpha]\right)}$$

$$\Bigg(-e^{2\,\pi^2\left(px^2+py^2\right)\lambda\,\sigma^2\,(px\,\text{Cos}[\beta]+py\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}\,\text{Cos}[2\,W0[px,\,py]]$$

$$\left(\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$

$$\left.\left.my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n\right)\sum_{n=1}^{M}-2\,\pi\,\text{Cos}\Big[$$

$$2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+my_n\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]$$

$$A_n\left(\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,mx_n\,\text{Tan}[\alpha]-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,my_n\,\text{Tan}[\alpha]\right)+$$

$$e^{4\,\pi^2\left(px^2+py^2\right)\lambda\,\sigma^2\,(px\,\text{Cos}[\beta]+py\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}\left(\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$

$$my_n\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\Big]\,A_n\Big)\sum_{n=1}^{M}-2\,\pi\,\mathrm{Sin}\Big[$$

$$2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+my_n\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\Big]$$

$$A_n\left(\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Sin}[\beta]\,mx_n\,\mathrm{Tan}[\alpha]-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Cos}[\beta]\,my_n\,\mathrm{Tan}[\alpha]\right)+$$

$$e^{2\,\pi^2\left(px^2+py^2\right)\lambda\,\sigma^2\,(px\,\mathrm{Cos}[\beta]+py\,\mathrm{Sin}[\beta])\,\mathrm{Tan}[\alpha]}\,\mathrm{Sin}[2\,W0[px,\,py]]$$

$$\left(\sum_{n=1}^{M}-\mathrm{Sin}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\right.\right.$$

$$my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\Big)\Big]\,A_n\Big)\sum_{n=1}^{M}-2\,\pi\,\mathrm{Sin}\Big[$$

$$2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+my_n\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\Big]$$

$$A_n\left(\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Sin}[\beta]\,mx_n\,\mathrm{Tan}[\alpha]-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Cos}[\beta]\,my_n\,\mathrm{Tan}[\alpha]\right)+$$

$$e^{2\,\pi^2\left(px^2+py^2\right)\lambda\,\sigma^2\,(px\,\mathrm{Cos}[\beta]+py\,\mathrm{Sin}[\beta])\,\mathrm{Tan}[\alpha]}\,\mathrm{Sin}[2\,W0[px,\,py]]$$

$$\left(\sum_{n=1}^{M}\mathrm{Cos}\Big[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\right.\right.$$

$$my_n\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\Big)\Big]\,A_n\Big)\sum_{n=1}^{M}-2\,\pi\,\mathrm{Cos}\Big[$$

$$2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\Big]$$

$$A_n\left(-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Sin}[\beta]\,mx_n\,\mathrm{Tan}[\alpha]+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Cos}[\beta]\,my_n\,\mathrm{Tan}[\alpha]\right)+$$

$$\left(\sum_{n=1}^{M}-\mathrm{Sin}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\right.\right.$$

$$my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\Big)\Big]\,A_n\Big)\sum_{n=1}^{M}-2\,\pi\,\mathrm{Cos}\Big[$$

$$2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\Big]$$

$$A_n\left(-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Sin}[\beta]\,mx_n\,\mathrm{Tan}[\alpha]+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Cos}[\beta]\,my_n\,\mathrm{Tan}[\alpha]\right)-$$

$$e^{2\,\pi^2\left(px^2+py^2\right)\lambda\,\sigma^2\,(px\,\mathrm{Cos}[\beta]+py\,\mathrm{Sin}[\beta])\,\mathrm{Tan}[\alpha]}\,\mathrm{Cos}[2\,W0[px,\,py]]$$

$$\left(\sum_{n=1}^{M}\mathrm{Cos}\Big[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\right.\right.$$

$$my_n\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\Big)\Big]\,A_n\Big)\sum_{n=1}^{M}-2\,\pi\,\mathrm{Sin}\Big[$$

$$2\,\pi\left(\mathrm{mx_n}\left(\mathrm{px}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\mathrm{my_n}\left(\mathrm{py}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\Big]$$

$$\mathrm{A_n}\left(-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{mx_n}\,\mathrm{Tan}[\alpha]+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{my_n}\,\mathrm{Tan}[\alpha]\right)+$$

$$\left(\sum_{\mathrm{n}=1}^{\mathrm{M}}\mathrm{Cos}\Big[2\,\pi\left(\mathrm{mx_n}\left(\mathrm{px}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\right.\right.$$

$$\left.\left.\mathrm{my_n}\left(\mathrm{py}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\Big]\,\mathrm{A_n}\right)$$

$$\left(-e^{2\,\pi^2\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\sigma^2\,(\mathrm{px}\,\mathrm{Cos}[\beta]+\mathrm{py}\,\mathrm{Sin}[\beta])\,\mathrm{Tan}[\alpha]}\,\mathrm{Sin}[2\,\mathrm{W0}[\mathrm{px},\,\mathrm{py}]]\right.$$

$$\sum_{\mathrm{n}=1}^{\mathrm{M}}-2\,\pi\,\mathrm{Cos}\Big[2\,\pi\left(\mathrm{mx_n}\left(\mathrm{px}-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+$$

$$\mathrm{my_n}\left(\mathrm{py}-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\Big]\,\mathrm{A_n}$$

$$\left(\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{mx_n}\,\mathrm{Tan}[\alpha]-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{my_n}\,\mathrm{Tan}[\alpha]\right)-$$

$$e^{2\,\pi^2\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\sigma^2\,(\mathrm{px}\,\mathrm{Cos}[\beta]+\mathrm{py}\,\mathrm{Sin}[\beta])\,\mathrm{Tan}[\alpha]}\,\mathrm{Cos}[2\,\mathrm{W0}[\mathrm{px},\,\mathrm{py}]]$$

$$\sum_{\mathrm{n}=1}^{\mathrm{M}}-2\,\pi\,\mathrm{Sin}\Big[2\,\pi\left(\mathrm{mx_n}\left(\mathrm{px}-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+$$

$$\mathrm{my_n}\left(\mathrm{py}-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\Big]\,\mathrm{A_n}$$

$$\left(\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{mx_n}\,\mathrm{Tan}[\alpha]-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{my_n}\,\mathrm{Tan}[\alpha]\right)+$$

$$\sum_{\mathrm{n}=1}^{\mathrm{M}}-2\,\pi\,\mathrm{Sin}\Big[2\,\pi\left(\mathrm{mx_n}\left(\mathrm{px}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+$$

$$\mathrm{my_n}\left(\mathrm{py}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\Big]\,\mathrm{A_n}$$

$$\left(-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{mx_n}\,\mathrm{Tan}[\alpha]+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{my_n}\,\mathrm{Tan}[\alpha]\right)\Big)+$$

$$e^{2\,\pi^2\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\sigma^2\,(\mathrm{px}\,\mathrm{Cos}[\beta]+\mathrm{py}\,\mathrm{Sin}[\beta])\,\mathrm{Tan}[\alpha]}\left(\sum_{\mathrm{n}=1}^{\mathrm{M}}-\mathrm{Sin}\Big[2\,\pi\left(\mathrm{mx_n}\left(\mathrm{px}-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\right.\right.$$

$$\left.\left.\mathrm{my_n}\left(\mathrm{py}-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\Big]\,\mathrm{A_n}\right)$$

$$\left(e^{2\,\pi^2\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\sigma^2\,(\mathrm{px}\,\mathrm{Cos}[\beta]+\mathrm{py}\,\mathrm{Sin}[\beta])\,\mathrm{Tan}[\alpha]}\sum_{\mathrm{n}=1}^{\mathrm{M}}-2\,\pi\,\mathrm{Cos}\Big[2\,\pi\left(\mathrm{mx_n}\left(\mathrm{px}-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\right.\right.\right.$$

$$\left.\left.\left.\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\mathrm{my_n}\left(\mathrm{py}-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\Big]\,\mathrm{A_n}$$

$$\left(\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{mx_n}\,\mathrm{Tan}[\alpha]-\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{my_n}\,\mathrm{Tan}[\alpha]\right)-$$

$$\text{Cos}[2\,\text{W0}[\text{px},\,\text{py}]]\sum_{n=1}^{M}-2\,\pi\,\text{Cos}\Big[2\,\pi\Big(\text{mx}_n\Big(\text{px}+\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\Big)+$$
$$\text{my}_n\Big(\text{py}+\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\Big)\Big)\Big]\,\text{A}_n$$
$$\Big(-\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Sin}[\beta]\,\text{mx}_n\,\text{Tan}[\alpha]+\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Cos}[\beta]\,\text{my}_n\,\text{Tan}[\alpha]\Big)-$$
$$\text{Sin}[2\,\text{W0}[\text{px},\,\text{py}]]\sum_{n=1}^{M}-2\,\pi\,\text{Sin}\Big[2\,\pi\Big(\text{mx}_n\Big(\text{px}+\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\Big)+$$
$$\text{my}_n\Big(\text{py}+\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\Big)\Big)\Big]\,\text{A}_n$$
$$\Big(-\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Sin}[\beta]\,\text{mx}_n\,\text{Tan}[\alpha]+\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Cos}[\beta]\,\text{my}_n\,\text{Tan}[\alpha]\Big)\Big)+$$
$$e^{4\,\pi^2\,(\text{px}^2+\text{py}^2)\,\lambda\,\sigma^2\,(\text{px}\,\text{Cos}[\beta]+\text{py}\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}\,\pi^2\,\text{px}^2\,\text{py}\,\lambda\,\sigma^2\,\text{Cos}[\beta]$$
$$\Big(\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\Big(\text{mx}_n\Big(\text{px}-\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\Big)+$$
$$\text{my}_n\Big(\text{py}-\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\Big)\Big)\Big]\,\text{A}_n\Big)^2$$
$$\text{Tan}[\alpha]+e^{4\,\pi^2\,(\text{px}^2+\text{py}^2)\,\lambda\,\sigma^2\,(\text{px}\,\text{Cos}[\beta]+\text{py}\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}\,\pi^2\,\text{py}^3\,\lambda\,\sigma^2\,\text{Cos}[\beta]$$
$$\Big(\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\Big(\text{mx}_n\Big(\text{px}-\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\Big)+$$
$$\text{my}_n\Big(\text{py}-\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\Big)\Big)\Big]\,\text{A}_n\Big)^2$$
$$\text{Tan}[\alpha]-e^{4\,\pi^2\,(\text{px}^2+\text{py}^2)\,\lambda\,\sigma^2\,(\text{px}\,\text{Cos}[\beta]+\text{py}\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}\,\pi^2\,\text{px}^3\,\lambda\,\sigma^2\,\text{Sin}[\beta]$$
$$\Big(\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\Big(\text{mx}_n\Big(\text{px}-\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\Big)+$$
$$\text{my}_n\Big(\text{py}-\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\Big)\Big)\Big]\,\text{A}_n\Big)^2$$
$$\text{Tan}[\alpha]-e^{4\,\pi^2\,(\text{px}^2+\text{py}^2)\,\lambda\,\sigma^2\,(\text{px}\,\text{Cos}[\beta]+\text{py}\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}\,\pi^2\,\text{px}\,\text{py}^2\,\lambda\,\sigma^2\,\text{Sin}[\beta]$$
$$\Big(\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\Big(\text{mx}_n\Big(\text{px}-\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\Big)+$$
$$\text{my}_n\Big(\text{py}-\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\Big)\Big)\Big]\,\text{A}_n\Big)^2\,\text{Tan}[\alpha]+\pi^2\,(\text{px}^2+\text{py}^2)\,\lambda\,\sigma^2$$
$$(-\text{py}\,\text{Cos}[\beta]+\text{px}\,\text{Sin}[\beta])\Big(\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\Big(\text{mx}_n\Big(\text{px}+\frac{1}{2}\,(\text{px}^2+\text{py}^2)\,\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\Big)+$$

$$\left. \left. \left. \

$$\text{Tan}[\alpha]\Big) + \text{my}_n\left(\text{py} - \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,\text{A}_n\Big)^2 +$$

$$\left(\sum_{n=1}^{M} -\text{Sin}\Big[2\,\pi\left(\text{mx}_n\left(\text{px} - \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) + \text{my}_n\right.\right.$$

$$\left.\left.\left(\text{py} - \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,\text{A}_n\right)^2\Bigg) +$$

$$\left(\sum_{n=1}^{M} -\text{Sin}\Big[2\,\pi\left(\text{mx}_n\left(\text{px} + \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) + \text{my}_n\left(\text{py} + \right.\right.\right.$$

$$\left.\left.\left.\frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,\text{A}_n\right)^2 -$$

$$2\,e^{2\,\pi^2\left(\text{px}^2+\text{py}^2\right)\lambda\,\sigma^2\,(\text{px}\,\text{Cos}[\beta]+\text{py}\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}\left(\left(\sum_{n=1}^{M} -\text{Sin}\Big[2\,\pi\left(\text{mx}_n\left(\text{px} - \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Cos}[\beta]\right.\right.\right.\right.$$

$$\left.\left.\left.\left.\text{Tan}[\alpha]\Big) + \text{my}_n\left(\text{py} - \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,\text{A}_n\right)\right.$$

$$\left(\text{Sin}\Big[\pi\left(\text{px}^2 + \text{py}^2\right)\lambda\left(-2\,\text{Z0} + \text{Cs}\left(\text{px}^2 + \text{py}^2\right)\lambda^2 + \text{ZA}\,\text{Sin}[2\,\chi - \chi 0]\right)\Big]\right.$$

$$\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(\text{mx}_n\left(\text{px} + \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) +$$

$$\text{my}_n\left(\text{py} + \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,\text{A}_n +$$

$$\text{Cos}\Big[\pi\left(\text{px}^2 + \text{py}^2\right)\lambda\left(-2\,\text{Z0} + \text{Cs}\left(\text{px}^2 + \text{py}^2\right)\lambda^2 + \text{ZA}\,\text{Sin}[2\,\chi - \chi 0]\right)\Big]$$

$$\sum_{n=1}^{M} -\text{Sin}\Big[2\,\pi\left(\text{mx}_n\left(\text{px} + \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) +$$

$$\left.\text{my}_n\left(\text{py} + \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,\text{A}_n\right) +$$

$$\left(\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(\text{mx}_n\left(\text{px} - \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) + \text{my}_n\right.\right.$$

$$\left.\left.\left(\text{py} - \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,\text{A}_n\right)$$

$$\left(\text{Cos}\Big[\pi\left(\text{px}^2 + \text{py}^2\right)\lambda\left(-2\,\text{Z0} + \text{Cs}\left(\text{px}^2 + \text{py}^2\right)\lambda^2 + \text{ZA}\,\text{Sin}[2\,\chi - \chi 0]\right)\Big]\right.$$

$$\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(\text{mx}_n\left(\text{px} + \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) +$$

$$\left.\text{my}_n\left(\text{py} + \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\Big)\Big]\,\text{A}_n -$$

$$\text{Sin}\left[\pi\left(\text{px}^2+\text{py}^2\right)\lambda\left(-2\,\text{Z0}+\text{Cs}\left(\text{px}^2+\text{py}^2\right)\lambda^2+\text{ZA Sin}[2\,\chi-\chi 0]\right)\right]$$

$$\sum_{n=1}^{M}-\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$

$$\left.\left.\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n\Bigg)\Bigg)$$

$$\left(-2\left(\text{px}^2+\text{py}^2\right)\lambda+2\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sec}[\alpha]^2+4\,(\text{px}\,\text{Cos}[\beta]+\text{py}\,\text{Sin}[\beta])\right.$$

$$\left.\text{Tan}[\alpha]\right)+$$

$$4\left(e^{4\pi^2\left(\text{px}^2+\text{py}^2\right)\lambda\,\sigma^2\,(\text{px}\,\text{Cos}[\beta]+\text{py}\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}\left(\left(\sum_{n=1}^{M}-\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.\right.\right.$$

$$\left.\left.\left.\left.\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n\right)$$

$$\sum_{n=1}^{M}-2\,\pi\,\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$

$$\left.\left.\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n$$

$$\left(-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\text{Cos}[\beta]\,\text{mx}_n\,\text{Tan}[\alpha]-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\text{Sin}[\beta]\,\text{my}_n\,\text{Tan}[\alpha]\right)+$$

$$\left(\sum_{n=1}^{M}\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.\right.$$

$$\left.\left.\left.\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n\right)$$

$$\sum_{n=1}^{M}-2\,\pi\,\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$

$$\left.\left.\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n$$

$$\left.\left(-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\text{Cos}[\beta]\,\text{mx}_n\,\text{Tan}[\alpha]-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\text{Sin}[\beta]\,\text{my}_n\,\text{Tan}[\alpha]\right)\right)+$$

$$\left(\sum_{n=1}^{M}-\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\right.\right.\right.$$

$$\left.\left.\left.\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n\right)$$

$$\sum_{n=1}^{M}-2\,\pi\,\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$

$$\left.\left.\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n$$

$$\left(\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\text{Cos}[\beta]\,\text{mx}_n\,\text{Tan}[\alpha]+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\text{Sin}[\beta]\,\text{my}_n\,\text{Tan}[\alpha]\right)+$$

$$\left(\sum_{n=1}^{M} \text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\right.\right.\right.$$
$$\left.\left.\left.\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n\right)$$
$$\sum_{n=1}^{M} -2\,\pi\,\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$
$$\left.\left.\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n$$
$$\left(\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\text{Cos}[\beta]\,\text{mx}_n\,\text{Tan}[\alpha]+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\text{Sin}[\beta]\,\text{my}_n\,\text{Tan}[\alpha]\right)-$$
$$e^{2\,\pi^2\left(\text{px}^2+\text{py}^2\right)\lambda\,\sigma^2\,(\text{px}\,\text{Cos}[\beta]+\text{py}\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}$$
$$\left(\left(\text{Sin}\left[\pi\left(\text{px}^2+\text{py}^2\right)\lambda\left(-2\,\text{Z0}+\text{Cs}\left(\text{px}^2+\text{py}^2\right)\lambda^2+\text{ZA}\,\text{Sin}[2\,\chi-\chi 0]\right)\right]\right.\right.$$
$$\sum_{n=1}^{M}\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$
$$\left.\left.\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n+$$
$$\text{Cos}\left[\pi\left(\text{px}^2+\text{py}^2\right)\lambda\left(-2\,\text{Z0}+\text{Cs}\left(\text{px}^2+\text{py}^2\right)\lambda^2+\text{ZA}\,\text{Sin}[2\,\chi-\chi 0]\right)\right]$$
$$\sum_{n=1}^{M}-\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$
$$\left.\left.\left.\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n\right)$$
$$\sum_{n=1}^{M}-2\,\pi\,\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$
$$\left.\left.\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n$$
$$\left(-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\text{Cos}[\beta]\,\text{mx}_n\,\text{Tan}[\alpha]-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\text{Sin}[\beta]\,\text{my}_n\,\text{Tan}[\alpha]\right)+$$
$$\left(\text{Cos}\left[\pi\left(\text{px}^2+\text{py}^2\right)\lambda\left(-2\,\text{Z0}+\text{Cs}\left(\text{px}^2+\text{py}^2\right)\lambda^2+\text{ZA}\,\text{Sin}[2\,\chi-\chi 0]\right)\right]\right.$$
$$\sum_{n=1}^{M}\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$
$$\left.\left.\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n-$$
$$\text{Sin}\left[\pi\left(\text{px}^2+\text{py}^2\right)\lambda\left(-2\,\text{Z0}+\text{Cs}\left(\text{px}^2+\text{py}^2\right)\lambda^2+\text{ZA}\,\text{Sin}[2\,\chi-\chi 0]\right)\right]$$
$$\sum_{n=1}^{M}-\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.$$

$$\left.\left.\left.

$$\left(\frac{1}{2}\left(px^2+py^2\right)\text{Cos}[\beta]\,mx_n\,\text{Tan}[\alpha]+\frac{1}{2}\left(px^2+py^2\right)\text{Sin}[\beta]\,my_n\,\text{Tan}[\alpha]\right)\Bigg)+$$

$$\left(\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+my_n\right.\right.$$

$$\left.\left.\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n\right)$$

$$\left(\pi\left(px^2+py^2\right)\text{Cos}\Big[\pi\left(px^2+py^2\right)\lambda\left(-2\,Z0+Cs\left(px^2+py^2\right)\lambda^2+ZA\,\text{Sin}[2\,\chi-\chi 0]\right)\Big]\right.$$

$$\left(-2\,Z0+3\,Cs\left(px^2+py^2\right)\lambda^2+ZA\,\text{Sin}[2\,\chi-\chi 0]\right)$$

$$\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.$$

$$\left.\left.my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n-$$

$$\pi\left(px^2+py^2\right)\left(-2\,Z0+3\,Cs\left(px^2+py^2\right)\lambda^2+ZA\,\text{Sin}[2\,\chi-\chi 0]\right)$$

$$\text{Sin}\Big[\pi\left(px^2+py^2\right)\lambda\left(-2\,Z0+Cs\left(px^2+py^2\right)\lambda^2+ZA\,\text{Sin}[2\,\chi-\chi 0]\right)\Big]$$

$$\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.$$

$$\left.\left.my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n+$$

$$\text{Cos}\Big[\pi\left(px^2+py^2\right)\lambda\left(-2\,Z0+Cs\left(px^2+py^2\right)\lambda^2+ZA\,\text{Sin}[2\,\chi-\chi 0]\right)\Big]$$

$$\sum_{n=1}^{M}-2\,\pi\,\text{Cos}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.$$

$$\left.\left.my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n$$

$$\left(\frac{1}{2}\left(px^2+py^2\right)\text{Cos}[\beta]\,mx_n\,\text{Tan}[\alpha]+\frac{1}{2}\left(px^2+py^2\right)\text{Sin}[\beta]\,my_n\,\text{Tan}[\alpha]\right)+$$

$$\text{Sin}\Big[\pi\left(px^2+py^2\right)\lambda\left(-2\,Z0+Cs\left(px^2+py^2\right)\lambda^2+ZA\,\text{Sin}[2\,\chi-\chi 0]\right)\Big]$$

$$\sum_{n=1}^{M}-2\,\pi\,\text{Sin}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.$$

$$\left.\left.my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,A_n$$

$$\left.\left.\left(\frac{1}{2}\left(px^2+py^2\right)\text{Cos}[\beta]\,mx_n\,\text{Tan}[\alpha]+\frac{1}{2}\left(px^2+py^2\right)\text{Sin}[\beta]\,my_n\,\text{Tan}[\alpha]\right)\right)\right)+$$

$$2\,e^{4\,\pi^2\left(px^2+py^2\right)\lambda\,\sigma^2\,(px\,\text{Cos}[\beta]+py\,\text{Sin}[\beta])\,\text{Tan}[\alpha]}\,\pi^2\left(px^2+py^2\right)$$

$$\sigma^2$$

$$(px\,\text{Cos}[\beta]+py\,\text{Sin}[\beta])$$

$$\left(\left(\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\right.\right.\right.$$

$$\left.\left.\left.

$$\mathrm{Sin}\left[\pi\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\left(-2\,\mathrm{Z0}+\mathrm{Cs}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda^2+\mathrm{ZA}\,\mathrm{Sin}[2\,\chi-\chi 0]\right)\right]$$
$$\sum_{n=1}^{M}-\mathrm{Sin}\left[2\,\pi\left(\mathrm{mx}_n\left(\mathrm{px}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Cos}[\beta]\,\mathrm{Tan}[\alpha]\right)+\right.\right.$$
$$\left.\left.\mathrm{my}_n\left(\mathrm{py}+\frac{1}{2}\left(\mathrm{px}^2+\mathrm{py}^2\right)\lambda\,\mathrm{Sin}[\beta]\,\mathrm{Tan}[\alpha]\right)\right)\Big]\,\mathrm{A}_n\right)\mathrm{Tan}[\alpha]\Bigg)\Bigg)$$

Derivative with respect to C_s:

$$\frac{\partial\,|Q(p)|^2}{\partial\,C_s} = -2\,e^{-\frac{1}{2}\pi^2\left(px^2+py^2\right)\sigma^2\left(-4+\left(px^2+py^2\right)\lambda^2-\left(px^2+py^2\right)\lambda^2\,Sec[\alpha]^2\right)}\,\pi\left(px^2+py^2\right)^2\lambda^3$$

$$\left(-\sum_{n=1}^{M}Cos\left[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,Cos[\beta]\,Tan[\alpha]\right)+my_n\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,Sin[\beta]\,Tan[\alpha]\right)\right)\right]\right.$$

$$A_n\left(Sin\left[\pi\left(px^2+py^2\right)\lambda\left(-2\,Z0+Cs\left(px^2+py^2\right)\lambda^2+ZA\,Sin[2\,\chi-\chi 0]\right)\right]\right.$$

$$\sum_{n=1}^{M}Cos\left[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,Cos[\beta]\,Tan[\alpha]\right)+\right.\right.$$

$$\left.\left.my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,Sin[\beta]\,Tan[\alpha]\right)\right)\right]A_n+$$

$$Cos\left[\pi\left(px^2+py^2\right)\lambda\left(-2\,Z0+Cs\left(px^2+py^2\right)\lambda^2+ZA\,Sin[2\,\chi-\chi 0]\right)\right]$$

$$\sum_{n=1}^{M}-Sin\left[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,Cos[\beta]\,Tan[\alpha]\right)+my_n\right.\right.$$

$$\left.\left.\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,Sin[\beta]\,Tan[\alpha]\right)\right)\right]A_n\Bigg)+$$

$$\left(\sum_{n=1}^{M}-Sin\left[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,Cos[\beta]\,Tan[\alpha]\right)+my_n\left(py-\right.\right.\right.\right.$$

$$\left.\left.\left.\left.\frac{1}{2}\left(px^2+py^2\right)\lambda\,Sin[\beta]\,Tan[\alpha]\right)\right)\right]A_n\right)$$

$$\left(Cos\left[\pi\left(px^2+py^2\right)\lambda\left(-2\,Z0+Cs\left(px^2+py^2\right)\lambda^2+ZA\,Sin[2\,\chi-\chi 0]\right)\right]\right.$$

$$\sum_{n=1}^{M}Cos\left[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,Cos[\beta]\,Tan[\alpha]\right)+\right.\right.$$

$$\left.\left.my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,Sin[\beta]\,Tan[\alpha]\right)\right)\right]A_n-$$

$$Sin\left[\pi\left(px^2+py^2\right)\lambda\left(-2\,Z0+Cs\left(px^2+py^2\right)\lambda^2+ZA\,Sin[2\,\chi-\chi 0]\right)\right]$$

$$\sum_{n=1}^{M}-Sin\left[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,Cos[\beta]\,Tan[\alpha]\right)+my_n\right.\right.$$

$$\left.\left.\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,Sin[\beta]\,Tan[\alpha]\right)\right)\right]A_n\Bigg)\Bigg) \tag{7}$$

Derivative with respect to Z_0:

$$\frac{\partial\,|Q(p)|^2}{\partial\,Z_0} = -4\,e^{-\frac{1}{2}\pi^2\left(px^2+py^2\right)\sigma^2\left(-4+\left(px^2+py^2\right)\lambda^2-\left(px^2+py^2\right)\lambda^2\,Sec[\alpha]^2\right)}\,\pi\left(px^2+py^2\right)\lambda$$

$$\left(\left(\sum_{n=1}^{M}Cos\left[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,Cos[\beta]\,Tan[\alpha]\right)+my_n\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,Sin[\beta]\,Tan[\alpha]\right)\right)\right]\right.\right.$$

$$A_n\Bigg)\Bigg(\text{Sin}\Big[\pi\left(px^2+py^2\right)\lambda\left(-2\,Z0+Cs\left(px^2+py^2\right)\lambda^2+ZA\,\text{Sin}[2\,\chi-\chi 0]\right)\Big]$$

$$\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]A_n+$$

$$\text{Cos}\Big[\pi\left(px^2+py^2\right)\lambda\left(-2\,Z0+Cs\left(px^2+py^2\right)\lambda^2+ZA\,\text{Sin}[2\,\chi-\chi 0]\right)\Big]$$

$$\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+my_n\right.$$

$$\left.\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]A_n\Bigg)+$$

$$\Bigg(\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+my_n\left(py-\right.\right.$$

$$\left.\left.\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]A_n\Bigg)$$

$$\Bigg(-\text{Cos}\Big[\pi\left(px^2+py^2\right)\lambda\left(-2\,Z0+Cs\left(px^2+py^2\right)\lambda^2+ZA\,\text{Sin}[2\,\chi-\chi 0]\right)\Big]$$

$$\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$my_n\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]A_n+$$

$$\text{Sin}\Big[\pi\left(px^2+py^2\right)\lambda\left(-2\,Z0+Cs\left(px^2+py^2\right)\lambda^2+ZA\,\text{Sin}[2\,\chi-\chi 0]\right)\Big]$$

$$\sum_{n=1}^{M}-\text{Sin}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+my_n\right.$$

$$\left.\left(py+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]A_n\Bigg)\Bigg)$$

Derivative with respect to Z_A:

$$\frac{\partial\,|Q(p)|^2}{\partial\,Z_A}=-2\,e^{-\frac{1}{2}\pi^2\left(px^2+py^2\right)\sigma^2\left(-4+\left(px^2+py^2\right)\lambda^2-\left(px^2+py^2\right)\lambda^2\,\text{Sec}[\alpha]^2\right)}\pi\left(px^2+py^2\right)\lambda\,\text{Sin}[2\,\chi-\chi 0]$$

$$\Bigg(-\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(mx_n\left(px-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+my_n\left(py-\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]$$

$$A_n\Bigg(\text{Sin}\Big[\pi\left(px^2+py^2\right)\lambda\left(-2\,Z0+Cs\left(px^2+py^2\right)\lambda^2+ZA\,\text{Sin}[2\,\chi-\chi 0]\right)\Big]$$

$$\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(mx_n\left(px+\frac{1}{2}\left(px^2+py^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+$$

$$\left.\left.\left.

$$\sum_{n=1}^{M} -\text{Sin}\Big[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,\text{A}_n\Bigg)+$$

$$\left(\sum_{n=1}^{M} -\text{Sin}\Big[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,\text{A}_n\right)$$

$$\Bigg(-\text{Cos}\Big[\pi\left(\text{px}^2+\text{py}^2\right)\lambda\left(-2\,\text{Z0}+\text{Cs}\left(\text{px}^2+\text{py}^2\right)\lambda^2+\text{ZA}\,\text{Sin}[2\,\chi-\chi 0]\right)\Big]$$

$$\sum_{n=1}^{M}\text{Cos}\Big[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,\text{A}_n+$$

$$\text{Sin}\Big[\pi\left(\text{px}^2+\text{py}^2\right)\lambda\left(-2\,\text{Z0}+\text{Cs}\left(\text{px}^2+\text{py}^2\right)\lambda^2+\text{ZA}\,\text{Sin}[2\,\chi-\chi 0]\right)\Big]$$

$$\sum_{n=1}^{M} -\text{Sin}\Big[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\Big]\,\text{A}_n\Bigg)\Bigg)$$

Appendix 2 : Fast Computation of the Derivatives

The derivatives in Appendix 1 are computed for an arbitrary spatial frequency chosen by the user by the IPLT algorithm called AnalyticalAbsQ (See Chapter 2). The algorithm precomputes a series of " atomic" terms that appear in identical form in several equations, and subtitutes the computed values when appropriate. This appendix describes the precomputed terms and shows how they are combined in the calculation of the intricate expressions presented in Appendix 1. This part of the manuscript has been created using the Mathematica program (Mathematica, 2009) : the same comments on equation numbering, mathematical notation and graphical layout made for Appendix 1 also apply here.

Precomputed terms :

$$\text{psq} = \text{px}\hat{}2 + \text{py}\hat{}2 \tag{1}$$

sumpluscos =

$$\sum_{n=1}^{M} \text{Cos}\left[2\pi\left(\text{mx}_n\left(\text{px} + \frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) + \text{my}_n\left(\text{py} + \frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right] \text{A}_n \tag{2}$$

sumplussin =

$$\sum_{n=1}^{M} -\text{Sin}\left[2\pi\left(\text{mx}_n\left(\text{px} + \frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) + \text{my}_n\left(\text{py} + \frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right] \text{A}_n \tag{3}$$

$$\text{sumplus2cos} = \sum_{n=1}^{M} -2\pi \text{Cos}\left[2\pi\left(\text{mx}_n\left(\text{px} + \frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) + \text{my}_n\left(\text{py} + \frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right] \left(\frac{1}{2}\text{mx}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Sec}[\alpha]^2 + \frac{1}{2}\text{my}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sec}[\alpha]^2\,\text{Sin}[\beta]\right)\text{A}_n \tag{4}$$

$$\text{sumplus2} = \sum_{n=1}^{M} -2\pi\left(\frac{1}{2}\text{mx}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Sec}[\alpha]^2 + \frac{1}{2}\text{my}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sec}[\alpha]^2\,\text{Sin}[\beta]\right) \text{Sin}\left[2\pi\left(\text{mx}_n\left(\text{px} + \frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) + \text{my}_n\left(\text{py} + \frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right] \text{A}_n \tag{5}$$

$$\text{sumplus3cos} = \sum_{n=1}^{M} -2\pi \text{Cos}\left[2\pi\left(\text{mx}_n\left(\text{px} + \frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) + \text{my}_n\left(\text{py} + \frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right] \text{A}_n\left(\frac{1}{2}\text{my}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha] - \frac{1}{2}\text{mx}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right) \tag{6}$$

$$\text{sumplus3sin} = \sum_{n=1}^{M} -2\pi \text{Sin}\left[2\pi\left(\text{mx}_n\left(\text{px} + \frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) + \text{my}_n\left(\text{py} + \frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right] \text{A}_n\left(\frac{1}{2}\text{my}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha] - \frac{1}{2}\text{mx}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right) \tag{7}$$

$$\text{sumplus4cos} = \sum_{n=1}^{M} -2\pi$$

$$\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]$$
$$\text{A}_n\left(\frac{1}{2}\,\text{mx}_n\left(\text{px}^2+\text{py}^2\right)\text{Cos}[\beta]\,\text{Tan}[\alpha]+\frac{1}{2}\,\text{my}_n\left(\text{px}^2+\text{py}^2\right)\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)$$

$$\text{sumplus4sin}=\sum_{n=1}^{M}-2\,\pi$$
$$\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}+\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right] \tag{9}$$
$$\text{A}_n\left(\frac{1}{2}\,\text{mx}_n\left(\text{px}^2+\text{py}^2\right)\text{Cos}[\beta]\,\text{Tan}[\alpha]+\frac{1}{2}\,\text{my}_n\left(\text{px}^2+\text{py}^2\right)\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)$$

$$\text{summinuscos}=$$
$$\sum_{n=1}^{M}\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n \tag{10}$$

$$\text{summinussin}=$$
$$\sum_{n=1}^{M}-\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n \tag{11}$$

$$\text{summinus2cos}=\sum_{n=1}^{M}-2\,\pi$$
$$\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right] \tag{12}$$
$$\left(-\frac{1}{2}\,\text{mx}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Sec}[\alpha]^2-\frac{1}{2}\,\text{my}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sec}[\alpha]^2\,\text{Sin}[\beta]\right)\text{A}_n$$

$$\text{summinus2}=\sum_{n=1}^{M}-2\,\pi\left(-\frac{1}{2}\,\text{mx}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Sec}[\alpha]^2-\frac{1}{2}\,\text{my}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sec}[\alpha]^2\,\text{Sin}[\beta]\right)$$
$$\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]\text{A}_n \tag{13}$$

$$\text{summinus3cos}=\sum_{n=1}^{M}-2\,\pi$$
$$\text{Cos}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right] \tag{14}$$
$$\text{A}_n\left(-\frac{1}{2}\,\text{my}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]+\frac{1}{2}\,\text{mx}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)$$

$$\text{summinus3sin}=\sum_{n=1}^{M}-2\,\pi$$
$$\text{Sin}\left[2\,\pi\left(\text{mx}_n\left(\text{px}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right)+\text{my}_n\left(\text{py}-\frac{1}{2}\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right] \tag{15}$$
$$\text{A}_n\left(-\frac{1}{2}\,\text{my}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]+\frac{1}{2}\,\text{mx}_n\left(\text{px}^2+\text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)$$

$$\text{summinus4cos} = \sum_{n=1}^{M} -2\pi$$
$$\text{Cos}\left[2\pi\left(\text{mx}_n\left(\text{px} - \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) + \text{my}_n\left(\text{py} - \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right]$$
$$\text{A}_n\left(-\frac{1}{2}\,\text{mx}_n\left(\text{px}^2 + \text{py}^2\right)\text{Cos}[\beta]\,\text{Tan}[\alpha] - \frac{1}{2}\,\text{my}_n\left(\text{px}^2 + \text{py}^2\right)\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)$$

$$\text{summinus4sin} = \sum_{n=1}^{M} -2\pi$$
$$\text{Sin}\left[2\pi\left(\text{mx}_n\left(\text{px} - \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Cos}[\beta]\,\text{Tan}[\alpha]\right) + \text{my}_n\left(\text{py} - \frac{1}{2}\left(\text{px}^2 + \text{py}^2\right)\lambda\,\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)\right)\right] \tag{17}$$
$$\text{A}_n\left(-\frac{1}{2}\,\text{mx}_n\left(\text{px}^2 + \text{py}^2\right)\text{Cos}[\beta]\,\text{Tan}[\alpha] - \frac{1}{2}\,\text{my}_n\left(\text{px}^2 + \text{py}^2\right)\text{Sin}[\beta]\,\text{Tan}[\alpha]\right)$$

$$\text{precomp1} = \left(\text{px}^2 + \text{py}^2\right)\lambda\,(\text{px}\,\text{Cos}[\beta] + \text{py}\,\text{Sin}[\beta])\,\sigma^2\,\text{Tan}[\alpha] \tag{18}$$

$$\text{precomp2} = e^{2\pi^2\left(\text{px}^2+\text{py}^2\right)\lambda\,(\text{px}\,\text{Cos}[\beta]+\text{py}\,\text{Sin}[\beta])\,\sigma^2\,\text{Tan}[\alpha]} \tag{19}$$

$$\text{precomp3} = e^{4\pi^2\left(\text{px}^2+\text{py}^2\right)\lambda\,(\text{px}\,\text{Cos}[\beta]+\text{py}\,\text{Sin}[\beta])\,\sigma^2\,\text{Tan}[\alpha]} \tag{20}$$

$$\text{precomp4minusplus} = \pi^2\left(\text{px}^2 + \text{py}^2\right)\lambda\,(-\text{py}\,\text{Cos}[\beta] + \text{px}\,\text{Sin}[\beta])\,\sigma^2 \tag{21}$$

$$\text{precomp4plusplus} = \pi^2\left(\text{px}^2 + \text{py}^2\right)(\text{px}\,\text{Cos}[\beta] + \text{py}\,\text{Sin}[\beta])\,\sigma^2 \tag{22}$$

$$\text{precomp5} = \pi\left(\text{px}^2 + \text{py}^2\right)\lambda\left(\left(-2\,\text{Z0} + \text{Cs}\left(\text{px}^2 + \text{py}^2\right)\lambda^2\right) + \text{ZA}\,\text{Sin}[2\,\chi - \chi 0]\right) \tag{23}$$

$$\text{precomp6} = \pi\left(\left(-2\,\text{Z0} + 3\,\text{Cs}\left(\text{px}^2 + \text{py}^2\right)\lambda^2\right) + \text{ZA}\,\text{Sin}[2\,\chi - \chi 0]\right) \tag{24}$$

$$\text{precomp7} = e^{-\frac{1}{2}\pi^2\,\text{psq}\,\sigma^2\left(4-\text{psq}\,\lambda^2+\text{psq}\,\lambda^2\,\text{Sec}[\alpha]^2+4\,\lambda\,(\text{px}\,\text{Cos}[\beta]+\text{py}\,\text{Sin}[\beta])\,\text{Tan}[\alpha]\right)} \tag{25}$$

$$\text{precomp8} = e^{\frac{1}{2}\pi^2\,\text{psq}\left(-4+\text{psq}\,\lambda^2-\text{psq}\,\lambda^2\,\text{Sec}[\alpha]^2\right)\sigma^2} \tag{26}$$

Eequation 2.25 in Chapter 2 :

$$\begin{aligned}|\text{Q}(\text{p})|^2 = \text{precomp7}\big((\text{sumpluscos})^2 + e^{4\pi^2\ \text{precomp1}}\left((\text{summinuscos})^2 + (\text{summinussin})^2\right) + (\text{sumplussin})^2 - \\ 2\,e^{2\pi^2\ \text{precomp1}}\,((\text{summinussin})\,(\text{Sin}[2\,\text{W0}[\text{px},\,\text{py}]]\,\text{sumpluscos} + \text{Cos}[2\,\text{W0}[\text{px},\,\text{py}]]\,\text{sumplussin}) + \\ (\text{summinuscos})\,(\text{Cos}[2\,\text{W0}[\text{px},\,\text{py}]]\,\text{sumpluscos} - \text{Sin}[2\,\text{W0}[\text{px},\,\text{py}]]\,\text{sumplussin}))\big)\end{aligned} \tag{27}$$

Derivative with respect to α :

$$\begin{aligned}\partial\,|\text{Q}(\text{p})|^2 \,/\, \partial\,\alpha = \frac{1}{2}\,\text{precomp7} \\ \big(4\,\big(2\,\text{precomp2}^2\,\pi^2\,\text{psq}\,\lambda\,\text{Sec}[\alpha]^2\,(\text{px}\,\text{Cos}[\beta] + \text{py}\,\text{Sin}[\beta])\,\sigma^2\left((\text{summinuscos})^2 + (\text{summinussin})^2\right) + \\ \text{precomp2}^2\,((\text{summinus2cos})\,\text{summinussin} + (\text{summinuscos})\,\text{summinus2}) + \\ (\text{sumplus2cos})\,\text{sumplussin} - 2\ \text{precomp2}\ \pi^2\,\text{psq}\ \lambda\,\text{Sec}[\alpha]^2\,(\text{px}\,\text{Cos}[\beta] + \text{py}\,\text{Sin}[\beta])\,\sigma^2 \\ ((\text{summinussin})\,(\text{Sin}[2\,\text{W0}[\text{px},\,\text{py}]]\ \text{sumpluscos} + \text{Cos}[2\,\text{W0}[\text{px},\,\text{py}]]\,\text{sumplussin}) + \\ (\text{summinuscos})\,(\text{Cos}[2\,\text{W0}[\text{px},\,\text{py}]]\ \text{sumpluscos} - \text{Sin}[2\,\text{W0}[\text{px},\,\text{py}]]\,\text{sumplussin})) + \\ (\text{sumpluscos})\,\text{sumplus2} - \text{precomp2} \\ ((\text{summinus2cos})\,(\text{Sin}[2\,\text{W0}[\text{px},\,\text{py}]]\,\text{sumpluscos} + \text{Cos}[2\,\text{W0}[\text{px},\,\text{py}]]\ \text{sumplussin}) + \\ (\text{summinus2})\,(\text{Cos}[2\,\text{W0}[\text{px},\,\text{py}]]\,\text{sumpluscos} - \text{Sin}[2\,\text{W0}[\text{px},\,\text{py}]]\ \text{sumplussin}) +\end{aligned} \tag{28}$$

(summinuscos) (−Sin[2 W0[px, py]] sumplus2cos + Cos[2 W0[px, py]] sumplus2) +
(summinussin) (Cos[2 W0[px, py]] sumplus2cos + Sin[2 W0[px, py]] sumplus2)))) −
$2\,\pi^2$ psq λ Sec[α]^2 σ^2 ((sumpluscos)^2 + precomp2^2 ((summinuscos)^2 + (summinussin)^2) +
(sumplussin)^2 − 2 precomp2
((summinussin) (Sin[2 W0[px, py]] sumpluscos + Cos[2 W0[px, py]] sumplussin) +
(summinuscos) (Cos[2 W0[px, py]] sumpluscos − Sin[2 W0[px, py]] sumplussin)))
(2 px Cos[β] + 2 py Sin[β] + psq λ Tan[α]))

Derivative with respect to β :

∂ |Q (p)|^2 / ∂ β =
2 precomp7 (−precomp2 Cos[2 W0[px, py]] (sumplussin) summinus3cos + precomp3 (summinuscos)
summinus3sin + precomp2 Sin[2 W0[px, py]] (sumplussin) summinus3sin +
precomp2 Sin[2 W0[px, py]] (summinuscos) sumplus3cos + (sumplussin) sumplus3cos −
precomp2 Cos[2 W0[px, py]] (summinuscos) sumplus3sin +
(sumpluscos) (−precomp2 Sin[2 W0[px, py]] summinus3cos −
precomp2 Cos[2 W0[px, py]] summinus3sin + sumplus3sin) +
precomp2 (summinussin) (precomp2 summinus3cos − Cos[2 W0[px, py]] sumplus3cos − (29)
Sin[2 W0[px, py]] sumplus3sin) + precomp3 π^2 px^2 py λ σ^2 Cos[β] (summinuscos)^2 Tan[α] +
precomp3 π^2 py^3 λ σ^2 Cos[β] (summinuscos)^2 Tan[α] − precomp3 π^2 px^3 λ σ^2 Sin[β]
(summinuscos)^2 Tan[α] − precomp3 π^2 px py^2 λ σ^2 Sin[β] (summinuscos)^2 Tan[α] +
π^2 (px^2 + py^2) λ σ^2 (−py Cos[β] + px Sin[β]) (sumpluscos)^2 Tan[α] −
precomp3 π^2 (px^2 + py^2) λ σ^2 (−py Cos[β] + px Sin[β]) (summinussin)^2 Tan[α] −
π^2 px^2 py λ σ^2 Cos[β] (sumplussin)^2 Tan[α] − π^2 py^3 λ σ^2 Cos[β] (sumplussin)^2 Tan[α] +
π^2 px^3 λ σ^2 Sin[β] (sumplussin)^2 Tan[α] + π^2 px py^2 λ σ^2 Sin[β] (sumplussin)^2 Tan[α])

Derivative with respect to λ :

∂ |Q (p)|^2 / ∂ λ = $\frac{1}{2}$ precomp7
(−π^2 psq σ^2 ((sumpluscos)^2 + precomp3 ((summinuscos)^2 + (summinussin)^2) + (sumplussin)^2 −
2 precomp2 ((summinussin) (Sin[precomp5] sumpluscos + Cos[precomp5] sumplussin) +
(summinuscos) (Cos[precomp5] sumpluscos − Sin[precomp5] sumplussin)))
(−2 psq λ + 2 psq λ Sec[α]^2 + 4 (px Cos[β] + py Sin[β]) Tan[α]) +
4 (precomp3 ((summinussin) summinus4cos + (summinuscos) summinus4sin) +
(sumplussin) sumplus4cos + (sumpluscos) sumplus4sin −
precomp2 ((Sin[precomp5] sumpluscos + Cos[precomp5] sumplussin) summinus4cos +
(Cos[precomp5] sumpluscos − Sin[precomp5] sumplussin) summinus4sin +
(summinuscos) (−psq (precomp6) Sin[precomp5] sumpluscos − (30)
psq Cos[precomp5] (precomp6) sumplussin − Sin[precomp5] sumplus4cos +
Cos[precomp5] sumplus4sin) + (summinussin)
(psq Cos[π ((px^2 + py^2) λ ((−2 Z0 + Cs (px^2 + py^2) λ^2) + ZA Sin[2 χ − χ0])))
(precomp6) sumpluscos − psq (precomp6) Sin[precomp5] sumplussin +
Cos[precomp5] sumplus4cos + Sin[precomp5] sumplus4sin)) +
2 precomp3 precomp4plusplus ((summinuscos)^2 + (summinussin)^2) Tan[α] −
2 precomp2 precomp4plusplus
((summinussin) (Sin[precomp5] sumpluscos + Cos[precomp5] sumplussin) +
(summinuscos) (Cos[precomp5] sumpluscos − Sin[precomp5] sumplussin)) Tan[α]))

Derivative with respect to C_s :

$$\partial\,|Q\,(p)|^2 \,/\, \partial\,C_s = \\ -2\;\text{precomp8}\;\text{psq}^2\,\lambda^3\,\pi\,(-\text{summinuscos}\,(\text{Sin}[\text{precomp5}]\,\text{sumpluscos} + \text{Cos}[\text{precomp5}]\,\text{sumplussin}) + \\ (\text{summinussin})\,(\text{Cos}[\text{precomp5}]\,\text{sumpluscos} - \text{Sin}[\text{precomp5}]\,\text{sumplussin})) \tag{31}$$

Derivative with respect to Z_0 :

$$\partial\,|Q\,(p)|^2 \,/\, \partial\,\chi_0 = \\ -4\,\text{precomp8}\;\text{psq}\;\lambda\,\pi\,((\text{summinuscos})\,(\text{Sin}[\text{precomp5}]\,\text{sumpluscos} + \text{Cos}[\text{precomp5}]\,\text{sumplussin}) + \\ (\text{summinussin})\,(-\text{Cos}[\text{precomp5}]\,\text{sumpluscos} + \text{Sin}[\text{precomp5}]\,\text{sumplussin})) \tag{32}$$

Derivative with respect to Z_A :

$$Q\,\partial\,|Q\,(p)|^2 \,/\, \partial\,Z_A = -2\;\text{precomp8}\;\pi\,\text{psq}\;\lambda\,\text{Sin}[2\,\chi - \chi 0] \\ (-\text{summinuscos}\,(\text{Sin}[\text{precomp5}]\,\text{sumpluscos} + \text{Cos}[\text{precomp5}]\,\text{sumplussin}) + \\ (\text{summinussin})\,(\text{Cos}[\text{precomp5}]\,\text{sumpluscos} - \text{Sin}[\text{precomp5}]\,\text{sumplussin})) \tag{33}$$

Derivative with respect to χ_0 :

$$\partial\,|Q\,(p)|^2 \,/\, \partial\,\chi_0 = -2\;\text{precomp8}\;\pi\,\text{psq}\;\text{ZA}\,\lambda\,\text{Cos}[2\,\chi - \chi 0] \\ ((\text{summinuscos})\,(\text{Sin}[\text{precomp5}]\,\text{sumpluscos} + \text{Cos}[\text{precomp5}]\,\text{sumplussin}) + \\ (\text{summinussin})\,(-\text{Cos}[\text{precomp5}]\,\text{sumpluscos} + \text{Sin}[\text{precomp5}]\,\text{sumplussin})) \tag{34}$$

Appendix 3 : Derivatives for the TCIF Inversion

The IPLT algorithm called TCIFFitt, described in Chapter 2, performs a forward fitting inversion of the TCIF using the Levenberg-Marquardt algorithm (Marquardt, 1963) . Internally, the algorithm needs to compute the derivatives of the TCIF model with respect to the unknown terms (which are the values of the pixels of the original sample function). The algorithm performs the calculation by precomputing the values of two summation terms. These values are then substitued where appropriate in more complex expressions. This section of the manuscript describes the precomputed summation terms and shows how they are combined in the calculation of the derivatives. The usual remarks about equation numbering, mathematical notation and graphical layout in the appendices all apply.

The precomputed summation terms are:

$$\mathrm{sinc1}(n) = \mathrm{Sinc}(\boldsymbol{p}_{\mathbf{nminus}} - \boldsymbol{p}_{\boldsymbol{n}}) \tag{1}$$

$$\mathrm{sinc2}(n) = \mathrm{Sinc}\left(\boldsymbol{p}_{\mathbf{nplus}} - \boldsymbol{p}_{\boldsymbol{n}}\right) \tag{2}$$

Where p_{minus}, p_{plus} and Sinc are defined in Equations (2.32),(2.33) and (2.34) in Chapter 2, and n is the index of a pixel in the image.

When Equation (2.37) of the same chapter is referred to a pixel with index n, it can be written in the following form:

$$Q_n = a_n + \iota\, b_n \tag{3}$$

The derivatives of the two terms with respect to the values of the amplitude and phase of the pixel n in the original sample function have then the form:

$$\partial\, \mathrm{a_n} / \partial\, \mathrm{A_n} = \mathrm{Sin}[(\mathrm{W_0}\,(\mathrm{p}) - \gamma_\mathrm{n})]\, \mathrm{sinc1}\,(\mathrm{n}) + \mathrm{Sin}[(\mathrm{W_0}\,(\mathrm{p}) + \gamma_\mathrm{n})]\, \mathrm{sinc2}\,(\mathrm{n}) \tag{4}$$

$$\partial\, \mathrm{a_n} / \partial\, \gamma_\mathrm{n} = -\mathrm{Cos}[(\mathrm{W_0}\,(\mathrm{p}) - \gamma_\mathrm{n})]\, \mathrm{A_n}\, \mathrm{sinc1}\,(\mathrm{n}) + \mathrm{Cos}[(\mathrm{W_0}\,(\mathrm{p}) + \gamma_\mathrm{n})]\, \mathrm{sinc2}\,(\mathrm{n}) \tag{5}$$

$$\partial\, \mathrm{b_n} / \partial\, \mathrm{A_n} = \mathrm{Cos}[(\mathrm{W_0}\,(\mathrm{p}) - \gamma_\mathrm{n})]\, \mathrm{sinc1}\,(\mathrm{n}) - \mathrm{Cos}[(\mathrm{W_0}\,(\mathrm{p}) + \gamma_\mathrm{n})]\, \mathrm{sinc2}\,(\mathrm{n}) \tag{6}$$

$$\partial\, \mathrm{b_n} / \partial \gamma_\mathrm{n} = \mathrm{Sin}[(\mathrm{W_0}\,(\mathrm{p}) - \gamma_\mathrm{n})]\, \mathrm{A_n}\, \mathrm{sinc1}\,(\mathrm{n}) + \mathrm{Sin}[(\mathrm{W_0}\,(\mathrm{p}) + \gamma_\mathrm{n})]\, \mathrm{sinc2}\,(\mathrm{n}) \tag{7}$$

References

Arfken, H. J. W. G. B. (2005). *Mathematical Methods For Physicists*. Academic Press, 6 edition.

Berman, H., Henrick, K. and Nakamura, H. (2003). Announcing the worldwide protein data bank. *Nat Struct Mol Biol*, 10(12):980.

Boothroyd, C. (1998). Why don't high-resolution simulations and images match?. *Journal of Microscopy*, 190(Pts 1/2):99–108.

Boothroyd, C. (2000). Quantification of high-resolution electron microscope images of amorphous carbon. *Ultramicroscopy*, pages 159–168.

Boothroyd, C. B. and Yeadon, M. (2003). The phonon contribution to high-resolution electron microscope images. *Ultramicroscopy*, 96(3-4):361–5.

Castano-Diez, D., Moser, D., Schoenegger, A., Pruggnaller, S. and Frangakis, A. S. (2008). Performance evaluation of image processing algorithms on the GPU. *Journal of Structural Biology*, 164(1):153–160.

Chacón, P. and Wriggers, W. (2002). Multi-resolution contour-based fitting of macromolecular structures. *Journal of Molecular Biology*, 317(3):375–84.

Collaborative Computational Project, Number 4 (1994). The CCP4 suite: programs for protein crystallography. *Acta Crystallographica. Section D, Biological Crystallography*, 50(Pt 5):760–3.

Cowley, J. M. (1992). *Electron Diffraction Techniques Vol 1 (International Union of Crystallography*. Oxford University Press, USA.

Crowter, R., Henderson, R. and Smith, J. (1996). MRC image processing programs. *Journal Of Structural Biology*, pages 9–16.

CTF Simulation (2009). *http://stahlberglab.org/download/ctf-simulation*.

Dorset, D. (1995). *Structural Electron Crystallography*. Springer, 1 edition.

Engel, A., Wiggins, J. and Woodruff, D. (1974). A comparison of calcuated images generated by six modes of transmission electron microscopy. *Journal of Applied Physics*, 45(6):2739–2747.

Frank, J. (1996). *Three-dimensional electron microscopy of macromolecular assemblies*.

Frank, J. (2006). *Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State*. Oxford University Press, USA, 2 edition.

Gnu Scientific Library (2009). *http://www.gnu.org/software/gsl/*.

Goodman, J. W. (2005). *Introduction to Fourier optics*.

Haase (1970). Zusammenstellung der koeffizienten für die anpassung komplexer streufaktoren für schnelle elektronen durch polynome. *Zeitschrift für Naturforschung, Teil A*, pages 1219–1235.

Hahn, T. (2002). *International Tables for Crystallography, Volume A: Space Group Symmetry*. Springer, 5th edition.

Henderson, R., Baldwin, J., Downing, K., Lepault, J. and Zemlin, F. (1986). Structure of purple membrane from halobacterium halobium: recording, measurement and evaluation of electron micrographs at 3.5 Å resolution.. *Ultramicroscopy*, pages 147–178.

Henderson, R., Baldwin, J. M., Ceska, T. A., Zemlin, F. and Beckmann, E.et al. (1949). Communication in the presence of noise. *Proceedings of IRE*, pages 10-21.

Henderson, R., Baldwin, J. M., Ceska, T. A., Zemlin, F. and Beckmann, E.et al. (1990). Model for the structure of bacteriorhodopsin based on high-resolution electron cryo-microscopy. *Journal of Molecular Biology*, 213(4):899–929.

Herring, R. (2006). Coherence measurements of zero-loss, plasmon-loss and phonon-loss electrons and their contribution to the stobbs factor. *Ultramicroscopy*, 106(10):960–961.

Heyd, J. and Birmanns, S. (2008). Beyond the black box: Interactive global docking of protein complexes. *Microscopy Today*, 16(4).

Heymann, J. B. and Belnap, D. M. (2007). Bsoft: image processing and molecular modeling for electron microscopy. *Journal of Structural Biology*, 157(1):3–18.

Holm, P. J., Bhakat, P., Jegerschoeld, C., Gyobu, N. and Mitsuoka, K.et al. (2006). Structural basis for detoxification and oxidative stress protection in membranes. *Journal of Molecular Biology*, 360(5):934–945.

Howie, A. (2004). Hunting the stobbs factor. *Ultramicroscopy*, 98(2-4):73–79.

Hÿtch, M. and Stobbs, W. (1994). Quantitative comparison of high resolution TEM images with image simulations. *Ultramicroscopy*, 53(3):191–203.

Jegershoeld, C., Pawelzik, S., Purhonen, P., Bhakat, P. and Gheorghe, K. R.et al. (2008). Structural basis for induced formation of the inflammatory mediator prostaglandin e2. *Proceedings of the National Academy of Sciences of the United States of America*, 105(32):11110–11115.

Kalarat, K., Narkbuakaew, W., Pintavirooj, C. and Sangworasil, M. (2005). Rapid simultaneous algebraic reconstruction technique (SART) for Cone-Beam geometry on clustering system. In *TENCON 2005 2005 IEEE Region 10*, pages 1–4.

Karney, C. F. F. (2007). Quaternions in molecular modeling. *Journal of Molecular Graphics & Modelling*, 25(5):595–604.

Katchalski-Katzir, E., Shariv, I., Eisenstein, M., Friesem, A. A. and Aflalo, C.et al. (1992). Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proceedings of the National Academy of Sciences of the United States of America*, 89(6):2195–9.

Marquardt, D. (1963). An algorithm for Least-Squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):441, 431.

Mathematica (2009). *http://www.wolfram.com/products/mathematica/index.html*.

Mindell, J. A. and Grigorieff, N. (2003). Accurate determination of local defocus and specimen tilt in electron microscopy. *Journal of Structural Biology*, 142(3):334–47.

Mitsuoka, K., Hirai, T., Murata, K., Miyazawa, A. and Kidera, A.et al. (1999). The structure of bacteriorhodopsin at 3.0 a resolution based on electron crystallography: implication of the charge distribution. *Journal of Molecular Biology*, 286(3):861–882.

Murata, K., Mitsuoka, K., Hirai, T., Walz, T. and Agre, P.et al. (2000). Structural determinants of water permeation through aquaporin-1. *Nature*, 407(6804):599–605.

Natterer, F. (1986). *The Mathematics of Computerized Tomography*. John Wiley & Sons Inc.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4):308–313.

Peng, L. M. (1999). Electron atomic scattering factors and scattering potentials of crystals. *Micron*, 30(6):625–648.

Peng, L. M., Ren, G., Dudarev, S. and Whelan, M. (1996). Robust parameterization of elastic and absorptive electron atomic scattering factors. *Acta Crystallographica Section A*, pages 257–276.

Philippsen, A., Engel, H. and Engel, A. (2006). The contrast-imaging function for tilted specimens. *Ultramicroscopy*, 107(2-3):202–12.

Philippsen, A., Schenk, A. D., Signorell, G. A., Mariani, V. and Berneche, S.et al. (2007). Collaborative EM image processing with the IPLT image processing library and toolbox. *Journal of Structural Biology*, 157(1):28–37.

Polak, E. (1971). *Computational methods in optimization; a unified approach*. Academic Press, New York.

Reimer, L. (1997). *Transmission Electron Microscopy. Physics of Image Formation and Microanalysis*. Springer-Verlag GmbH, 4. A. edition.

Ren, G., Cheng, A., Reddy, V., Melnyk, P. and Mitra, A. K. (2000). Three-dimensional fold of the human AQP1 water channel determined at 4 a resolution by electron crystallography of two-dimensional crystals embedded in ice. *Journal of Molecular Biology*, 301(2):369–387.

Schafer, L., Yates, A. and Bonham, R. (1971). New values for the partial wave electron scattering factor for the elements 1<=Z<=57 and 72<=Z<=90 for incident electron energies of 10,40,70, and 100 keV. *Journal of Chemical Physics*, 55(6):3055–2056.

Schenk, A., Mariani, V., Engel, A. and Philippsen, A. (2009). Improved unbending for 2D electron crystallography. *In preparation*.

Shaikh, T. R., Gao, H., Baxter, W. T., Asturias, F. J. and Boisset, N.et al. (2008). SPIDER image processing for single-particle reconstruction of biological macromolecules from electron micrographs. *Nat. Protocols*, 3(12):1941–1974.

Sorzano, C. O. S., Jonic, S., Núñez-Ramírez, R., Boisset, N. and Carazo, J. M. (2007). Fast, robust, and accurate determination of transmission electron microscopy contrast transfer function. *Journal of Structural Biology*, 160(2):249–62.

Tang, G., Peng, L., Baldwin, P. R., Mann, D. S. and Jiang, W.et al. (2007). EMAN2: an extensible image processing suite for electron microscopy. *Journal of Structural Biology*, 157(1):38–46.

Thust, A. (2008). *The Stobbs factor in HRTEM: Hunt for a phantom?*, pages 163–164.

Vetterling, W. T. and Flannery, B. P. (2002). *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, 2 edition.

Volkmann, N. and Hanein, D. (1998). Quantitative fitting of atomic models into observed densities derived by electron microscopy. *Journal of Structural Biology*, 125(2-3):176–84.

Wriggers, W. and Chacon, P. (2001). Modeling tricks and fitting techniques for multiresolution structures. *Structure*, 9(9):779–788.

wxWidgets (2009). *http://www.wxwigets.org*.

Young, I. T. and van Vliet, L. J. (1995). Recursive implementation of the gaussian filter. *Signal Process.*, 44(2):139–151.

Zhou, Z. H. (2008). Towards atomic resolution structural determination by single-particle cryo-electron microscopy. *Current Opinion in Structural Biology*, 18(2):218–28.

Valerio Mariani

Biographical Information

First Name:	Valerio
Last Name:	Mariani
Birth Date:	March 28th, 1973
Birthplace:	Monza (Milano) – Italy
Sex:	Male

Current Position

Since 2009 — Biozentrum, University of Basel & SIB Swiss Institute of Bioinformatics

Post-Doctoral Research Associate on integrative modeling at the Protein Structure Bioinformatics Group (Schwede Lab)

Education

2005-2009 — M.E. Müller Institute for Microscopy, Biozentrum, University of Basel

- PhD - Bio-physics
- PI: Andreas Engel, PhD
- RESEARCH WORK:
  - Simulation and analysis of contrast transfer function for tilted samples in electron microscopy
  - Developer of the IPLT software package for image processing in 2D electron crystallography (www.iplt.org)
- DISSERTATION: Transfer of tilted sample information in transmission electon microscopy

2004-2005 — University of Texas - Health Science Center - Houston

- School of Health Information Sciences
- Graduate student in the PhD program (predoctoral track)

- PI: Willy Wriggers, PhD
- GPA: 4.0 / 4.0
- RESEARCH WORK: Image processing of heterogeneous single particle images from cryo-EM
- RESEARCH WORK: Combining cryo-EM and X-ray crystallography data for multiresolution modeling

1997-2003 Universita' di Milano

- Milano, Italy
- “Laurea” degree in chemistry
- Five years course
- Grade: 110 / 110
- SPECIALTY: structural chemistry
- PI: Prof. Salvatore Lanzavecchia
- RESEARCH WORK: Development of software for multi-resolution modeling of large macromolecular assemblies combining data from electron microscopy and x-ray crystallography
- DISSERTATION: Docking of high resolution protein structures into 3D density distributions of the corresponding quaternary structure obtained from electron microscopy (In Italian)

1987-1992 Collegio Arciverscovile Ballerini

- Seregno (Milano), Italy
- “Maturita' scientifica” (High-school diploma)
- Five years course
- Grade 54 / 60

Work experience

2004-2005 Biomachina Lab, University of Texas Health Science Center Houston

- Graduate Research Assistant

- PI: Willy Wriggers, PhD
- RESEARCH AREAS:
  - Multi-resolution modeling of protein structures
  - Developer of the Situs docking software package

2003-2004 Salvatore's Lanzavecchia's lab at the Universita' di Milano,

- Milano, Italy
- Graduate Research Assistant
- PI: Prof. Salvatore Lanzavecchia
- RESEARCH AREAS:
  - CTF correction on particle images from metal-shadowing EM techniques.
  - Classification of particle images coming from cryo-EM experiments for further reconstruction processing.

Publications

- Philippsen A, Schenk AD, Signorell GA, Mariani V, Berneche S, Engel A. Collaborative EM image processing with the IPLT image processing library and toolbox. *J Struct Biol.* 2007 Jan;157(1):28-37.

Awards

Keck Fellowship - Nanobiology Training Program (2005)

Paul C. Boyle Award for Excellence in Student Research (2005)

Additional work experience

1995-1996 IBM Corp

- Vimercate plant, Milano, Italy
- Quality Control
- Thermal and electrical tests on CPUs and memory cards

Additional background

- Experience in UNIX/LINUX system administration
- Proficient in C, C++ programming
- Proficient in electron microscopy image processing packages (IPLT, Situs, Bsoft, Xmipp, EMAN)
- Experience with LEO 912 and Philiphs CM 200 FEG electron microscope